***چشم‌انداز مدیریت صنعتی***
شماره ۱۷ ـ بهار ۱۳۹۴
ص‌ص ۱۱۸ ° ۹۹

# طراحی الگوی فرآیندی توسعه محصولات نرم‌افزاری با رویکرد مدل‌سازی ساختاری تفسیری

**محمدجواد نائیجی***

**چکیده**

ادبیات موجود درزمینه توسعه محصول جدید بسیار به محصولات فیزیکی توجه کرده است و این در حالی است که محصولات نرم‌افزاری ویژگی‌های منحصربه‌فردی دارند که باید به آن‌ها در فرآیند توسعه محصول توجه شود. هدف این مقاله شناسایی مراحل توسعه محصولات نرم‌افزاری و تحلیل تعاملات میان این مراحل، با استفاده از رویکرد مدل‌سازی ساختاری تفسیری است. با مرور ادبیات و تحلیل عاملی اکتشافی، هشت مرحله توسعه محصولات نرم‌افزاری، شامل تحلیل فرصت‌های بازار، ایده‌پردازی، ایجاد مفهوم، توسعه طراحی، مهندسی نمونه‌ای، بازبینی تولید، تولید انبوه و توسعه نسخه‌های اصلاحی شناسایی شدند. سپس این مراحل براساس قدرت وابستگی و قدرت نفوذ طبقه‌بندی شده است. مدل پیشنهادی این تحقیق به شناخت روابط و وابستگی مراحل توسعه محصول کمک می‌کند و نقشه راهی ارائه می‌دهد که می‌تواند احتمال موفقیت توسعه محصولات نرم‌افزاری را افزایش دهد.

**کلیدواژه‌ها: توسعه محصول جدید؛ توسعه نرم‌افزار؛ الگوی فرآیندی؛ مدل‌سازی ساختاری تفسیری.**

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی

---

تاریخ دریافت مقاله:۹۳/۱۲/۲۱، تاریخ پذیرش مقاله: ۹۴/۳/۱۰.

* استادیار، دانشگاه شهید بهشتی (نویسنده مسئول).

E-mail: M_naeiji@sbu.ac.ir

## ۱. مقدمه

با مرور ادبیات توسعه محصول جدید، دو جریان پژوهشی قابل شناسایی است. جریان نخست، به شناسایی عوامل مؤثر بر موفقیت توسعه محصول مربوط است که در مطالعات مختلف، گستره وسیعی از عوامل، مانند فرهنگ سازمانی، جو حاکم بر پروژه، قابلیت‌های تجاری‌سازی محصول، راهبرد [۶]، مدیریت پروژه، رهبری تیمی، طراحی مجدد سازمان [۵]، مشارکت مشتریان در فرآیند توسعه محصول [۲۰] و انعطاف‌پذیری محیط کار [۱۰] را شناسایی کرده است. جریان پژوهشی دوم، به‌دنبال تبیین فرآیندها و شناسایی ابعاد و مراحل مختلفی است که در توسعه یک محصول جدید رخ می‌دهد. این رویکرد که درپی ارائه الگوهای فرآیندی برای توسعه محصول است، بیش از ۶۰ سال سابقه دارد و تاکنون براساس آن، ده‌ها فرآیند مختلف برای توسعه محصولات طراحی و پیشنهاد شده است.

بیشتر این فرآیندها بر توسعه محصولات فیزیکی تمرکز دارند و به محصولات نرم‌افزاری توجه اندکی شده است. در همین مطالعات اندک نیز رویکرد غالب، معرفی فرآیندهای نظری است که توسط روش‌های کمی و کیفی آزمون نشده‌اند [۱۱]؛ درحالی که محصولات نرم‌افزاری دارای ویژگی‌هایی هستند که می‌توانند توسعه آن‌ها را از محصولات فیزیکی متمایز کنند. ویژگی نخست این است که محصولات نرم‌افزاری به بستر سخت‌افزاری نیاز دارند و باید با استانداردهای نرم‌افزاری و سخت‌افزاری موجود سازگار باشند. همچنین، این نرم‌افزارها به‌شدت در معرض خطر کپی‌برداری هستند و باید به‌نحوی طراحی شوند که بتوان از حقوق مالکیت فکری آن‌ها حمایت کرد [۲۱]. تولید انبوه محصولات نرم‌افزاری هزینه بسیار کمی دارد؛ اما هزینه‌های توسعه محصول و بازاریابی آن‌ها زیاد است. درنهایت، مشتریان انتظار دارند نرم‌افزارها به‌سرعت ارتقا یابند؛ اما بابت نسخه‌های جدید، افزایش قیمت زیاد را نمی‌پذیرند. در مدل‌های فرآیندی موجود در زمینه توسعه محصول، بسیاری از این الزامات درنظر گرفته نشده‌است که سبب عدم‌قابلیت تبیین‌کنندگی آن‌ها، در فرآیندهای واقعی توسعه محصولات نرم‌افزاری شده است.

برای تشریح بیشتر این شکاف، اشاره به دو نقطه‌ضعف مدل‌های موجود می‌تواند مفید باشد. نقطه ضعف نخست، بی‌توجهی به طراحی و ساخت واحدبه‌واحد محصولات نرم‌افزاری و سپس، یکپارچه‌سازی این واحدها است؛ به عبارت دیگر، برخلاف بیشتر محصولات فیزیکی، محصولات نرم‌افزاری در یک قالب کلی طراحی و ساخته نمی‌شوند [۲و۲۶]. همچنین، سازگاری نرم‌افزار با سخت‌افزارهایی که روی آن‌ها عمل می‌کند، از الزامات توسعه هر محصول نرم‌افزاری است که به‌کارگیری معماری منحصربه‌فردی را ایجاب می‌کند که باید در فرآیند توسعه محصول درنظر گرفته شود. نقطه‌ضعف دوم این است که در مدل‌های فعلی، فرایند توسعه محصول با عرضه انبوه محصول به بازار پایان می‌یابد و در برخی مدل‌ها، ارزیابی‌های پس از عرضه بازاری محصول،

به‌صورت یک حلقه بازخور، به ابتدای فرآیند توسعه محصول بازمی‌گردد [۱]. این در حالی است که بسیاری از مشکلات محصولات نرم‌افزاری، تنها با کاربرد وسیع مشخص می‌شود و ارائه نسخه‌های جدید به بازار، جزو طبیعی فرآیند توسعه محصول فعلی محسوب می‌شود و نمی‌توان ارائه این نسخه‌های جدید را بخش ابتدایی فرآیند توسعه محصولات جدید دانست.

با توجه به شکاف نظری فوق و الزامات متفاوتی که در توسعه محصولات نرم‌افزاری وجود دارد، در مقاله حاضر، با مرور ادبیات و مطالعه اکتشافی براساس نظرات خبرگان، یک مدل فرآیندی براساس روش معادلات ساختاری تفسیری (ISM) ارائه می‌شود. رویکرد ISM نقاط قوت بسیاری، مانند امکان تشریح الگوی روابط پیچیده میان مراحل یا متغیرها دارد [۴] که با به‌کارگیری آن، بهبود قابل توجهی در طراحی فرآیند توسعه محصول ایجاد می‌شود. استفاده از این رویکرد که در مطالعات توسعه محصول کمتر سابقه دارد، کمک می‌کند علاوه‌بر ترسیم مدل فرآیندی برای ابعاد و شاخص‌های توسعه محصول، سطوح مختلف اثرگذاری این ابعاد و شاخص‌ها را نیز تعیین کنیم.

## ۲. مبانی نظری و پیشینه پژوهش

برای توسعه اثربخش محصولات، به مدیران توصیه می‌شود از رویکردهای گام‌به‌گام و سلسله‌مراتبی استفاده کنند [۱۲] و وظایف، توالی و روابط موجود در هر گام را با دقت مشخص نمایند [۳۱] تا از این طریق، مخاطرات ناشی از توسعه محصولات جدید را کاهش و احتمال موفقیت نهایی محصول در بازار را افزایش دهند. همچنین، به‌کارگیری این مدل‌های فرآیندی، با ارائه یک زبان و چارچوب مشترک، اقدامات لازم برای پیاده‌سازی موفق توسعه محصول را آسان می‌کند [۱۵].

علیرغم تأکید بر اهمیت و اثربخشی رویکرد فرآیندی در توسعه محصولات جدید، با مرور پژوهش‌های متعددی که درخصوص مدل‌های فرآیندی توسعه محصول جدید صورت گرفته است، به‌سختی می‌توان مدل‌های منحصربه‌فردی یافت که اجتماع کلی درمورد آن‌ها وجود داشته باشد و هر پژوهشگر، متناسب با نوع صنعت [۱۳]، میزان رقابت در صنعت [۲۲] و نوع محصول، مراحل خاصی را برای فرآیند توسعه محصول پیشنهاد می‌کند. تحلیل عمیق‌تر چرایی تفاوت میان مدل‌ها، بیانگر وجود تعارض در پیش‌فرض مدل‌ها است که یکی از مهم‌ترین پیش‌فرض‌ها، اهمیت نقش و گستره متغیرهای واردشده به فرآیند توسعه محصول است. برای مثال، بارزاک و کان (۲۰۱۲) تنها ۱۵ درصد از موفقیت توسعه محصول را به طی کردن فرآیند رسمی توسعه مربوط دانسته‌اند و عواملی مانند راهبرد، تحقیقات بازار و تجاری‌سازی را مهم‌تر از فرآیند توصیف کرده‌اند [۶]؛ درحالی که برخی پژوهشگران دیگر، تمام موفقیت توسعه محصول جدید را به

فرآیند مربوط می‌دانند و تأکید می‌کنند که باید به فرآیند توسعه محصول آنقدر فراگیر و جامع نگریسته شود که سایر متغیرهای مؤثر بر موفقیت فرآیند، مانند هماهنگی محصول جدید با راهبردهای سازمان و نیازهای بازار، به‌عنوان جزئی از فرآیند توسعه دیده شوند [۱۸].

در تحقیق حاضر، برای طراحی فرآیند توسعه محصول، از پیش‌فرض دوم (نگرش فراگیر) استفاده شده است که اجازه می‌دهد مدل جامع‌تری ارائه شود. در این راستا، از چهار مدل اصلی و چهار مدل فرعی به‌عنوان پایه استخراج مدل‌ها و شاخص‌های توسعه محصول استفاده می‌شود. زوکاس و همکاران (۲۰۰۴)، کریشنان و پرابهو (۲۰۰۲)، لوه و همکاران (۲۰۰۹) و میسرا و همکاران (۲۰۱۲) از مدل‌های اصلی به‌کاررفته در این پژوهش هستند و سایر مدل‌هایی که در طراحی فرآیند توسعه محصول مورد توجه قرار گرفته‌اند، در جدول ۱ آمده‌اند.

یکی از مدل‌های فرآیندی بنیادین در توسعه محصول، مدل زوکاس و همکاران (۲۰۰۴) است [۳۶] که نوآوری عمده آن، جداسازی مراحل توسعه و دریچه‌های ارزیابی است. از دیدگاه این مطالعه، برای توسعه هر محصول، مراحلی طی می‌شود که پس از هر مرحله، یک دریچه ارزیابی برای ورود به مرحله گشوده می‌شود و تنها درصورت مثبت بودن نتیجه ارزیابی، امکان ورود به مرحله بعد وجود دارد. خلق ایده، توسعه محصول، ساخت قالب تجاری و راه‌اندازی تجاری از مراحل توسعه محصول و غربالگری ایده، آزمون مفهوم، تحلیل کسب‌وکار، آزمون محصول، تحلیل نتایج آزمون بازاری محصول و ارزیابی کوتاه‌مدت و بلندمدت پس از راه‌اندازی از دریچه‌های ارزیابی محصول هستند. کار ارزشمند دیگر این مقاله، طبقه‌بندی ابعاد و معیارهای ارزیابی توسعه محصول در پنج حوزه مختلف، شامل ارزیابی‌های بازار، مالی، محصول، فرآیند و شهود، است.

برای سازگار کردن فضای تحقیق با صنعت محصولات نرم‌افزاری، ابعاد و شاخص‌هایی از تحقیق کریشنان و پرابهو (۲۰۰۲) استخراج شده است [۲۱]. در این مطالعه، شش پروژه توسعه محصولات نرم‌افزاری در شش شرکت هندی با بافت‌های صنعتی، بازارها و سطوح متفاوت سفارشی‌سازی، بررسی شده است و ابعادی مانند تعریف و موضع‌یابی محصول، انتخاب محصول، فناوری، معماری و سفارشی‌سازی محصول، مدیریت نسخه‌های نرم‌افزار، بازاریابی، راه‌اندازی محصول، سیستم‌های پاداش و تشویق و سیستم‌های کیفیت، مورد شناسایی قرار گرفته است.

یکی از رویکردهای غالب در توسعه نرم‌افزارها که در سال‌های اخیر، مطالعات زیادی برمبنای آن صورت گرفته است، توسعه‌ی چابک نرم‌افزارها است [۳و۹و۱۹و۲۸] که پیش از آنکه روش خاصی را تجویز کند، اصولی را ارائه می‌کند که باید در همه روش‌ها رعایت شوند. در این تحقیق، تلاش شده است برای شناسایی مراحل توسعه نرم‌افزار، از اصول توسعه چابک استفاده شود. این اصول عبارت‌اند از [۱۴و ۲۷] : ۱. اولویت قرار دادن رضایت مشتری از راه ارائه

نرم‌افزاری باارزش؛ ۲. استقبال از هرگونه تغییری که به بهبود نرم‌افزار کمک می‌کند (حتی اگر در مراحل پایانی توسعه محصول باشد)؛ ۳. نرم‌افزار درحال تدوین برای دو هفته تا دو ماه به‌صورت آزمایشی به بازار عرضه شود؛ ۴. توسعه‌دهندگان فنی و تجاری، درطول پروژه، همیشه با هم ارتباط داشته باشند؛ ۵. محیطی انگیزشی برای افراد درگیر در پروژه فراهم شود؛ ۶. کاراترین و اثربخش‌ترین روش ارتباطی، تعامل چهره‌به‌چهره افراد است؛ ۷. باور به اینکه نرم‌افزار درحال ساخت، معیاری از پیشرفت کل شرکت است؛ ۸. باور به اینکه فرآیندهای چابک، توسعه پایدار را به‌همراه دارد؛ ۹. باور به اینکه توجه مستمر به تعالی فنی و طراحی خوب، چابکی را افزایش می‌دهد؛ ۱۰. باور به ضرورت ساده‌سازی؛ ۱۱. باور به اینکه بهترین معماری‌ها، الزامات و طرح‌ها، از تیم‌های خودسامان به‌دست می‌آید و ۱۲. باور به اینکه موفقیت ازطریق ارزیابی منظم میزان اثربخشی تیم توسعه نرم‌افزار و انطباق رفتارها با این معیارهای اثربخشی به‌دست می‌آید.

از نظر روش‌شناسی، مطالعه لوه و همکاران (۲۰۰۹)، بیشترین شباهت را به تحقیق حاضر دارد [۲۳]. در این مطالعه، از ISM به‌عنوان یکی از روش‌ها برای تدوین فرآیند توسعه محصول استفاده شده است و در این جهت، هفت مرحله اصلی، شامل برنامه محصول، خلق مفهوم، ارزیابی پروژه، توسعه طراحی، مهندسی معکوس، بازبینی تولید و تولید انبوه، شناسایی شده است. ۲۲ مرحله فرعی این مطالعه شامل برنامه‌ریزی تولید، تحلیل بازار، طراحی مفهومی، طراحی صنعتی، ارزیابی مفهوم، تحلیل هزینه، فروش و سود، ویژگی‌های محصول، طرح اولیه اجزاء، طراحی فنی، طراحی بسته‌بندی، طراحی الکترونیکی، طراحی مدارهای یکپارچه IC، طراحی نرم‌افزاری، تهیه نمونه اولیه، طراحی قالب، تعیین معیارهای بازبینی، تجهیز تولید، راه‌اندازی طرح مقدماتی، راه‌اندازی محصول مقدماتی، گرفتن تأیید نهایی از بازار و تولید انبوه است.

در جدول ۱، به برخی از مرتبط‌ترین مطالعات دیگر درزمینه فرآیند توسعه محصول اشاره شده است که برای استخراج ابعاد و شاخص‌های توسعه محصولات نرم‌افزاری به‌کار رفته‌اند.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی

*جدول ۱. مروری بر سایر تحقیقات به‌کاررفته برای شناسایی مراحل توسعه محصول*

| نویسنده | هدف تحقیق | مراحل (ابعاد) شناسایی‌شده فرآیند |
|---|---|---|
| رویس (۱۹۷۰) [۳۳] | بهبود روش‌های پیشین توسعه محصولات نرم‌افزاری و ارائه مدل آبشاری[1] | تهیه سند الزامات محصول، طراحی درقالب معماری نرم‌افزار، ساخت درقالب کدنویسی نرم‌افزار و ارائه محصول واقعی، یکپارچه‌سازی، آزمون و اشکال‌زدایی، راه‌اندازی، حفظ و نگهداری |
| بوهم (۱۹۸۸) [۷] | بهبود مدل آبشاری ازطریق درنظر گرفتن مخاطرات توسعه نرم‌افزار در هر مرحله از چرخه عمر و ارائه مدل حلزونی[2] | حلقه نخست: برنامه الزامات، برنامه چرخه عمر، مفهوم عملیاتی<br>حلقه دوم: برنامه توسعه، اعتبارسنجی الزامات<br>حلقه سوم: برنامه یکپارچه‌سازی و آزمایش، اعتبارسنجی طرح، طراحی محصول نرم‌افزاری<br>حلقه چهارم: اجرا، آزمون پذیرش، یکپارچه‌سازی و آزمایش، آزمون کدها، طراحی جزئی، تهیه نمونه عملیاتی<br>در هر حلقه، تحلیل مخاطره‌ها و ساخت نمونه اولیه مربوط به آن حلقه نیز به مراحل فرعی فوق اضافه می‌شود |
| زوکاس و همکاران (۲۰۰۴) [۳۶] | ارائه شواهد تجربی از معیارهای تجربی به‌کاررفته توسط مدیران باتجربه در امر توسعه محصول در انگلستان و هلند برای کنترل عملکرد در هریک از مراحل مختلف فرآیند توسعه محصول | شکل‌گیری ایده، غربالگری ایده، شکل‌گیری مفهوم، آزمون مفهوم، ساخت یک قالب کسب‌وکار، تحلیل کسب‌وکار، توسعه محصول، آزمون محصول، آزمون بازار، تحلیل نتایج آزمون بازار، راه‌اندازی محصول در بازار، ارزیابی‌های بعد از راه‌اندازی (کوتاه‌مدت)، ارزیابی‌های بعد از راه‌اندازی (بلندمدت) |
| هارمانجی‌اوغلو و همکاران (۲۰۰۷) [۱۸] | پاسخ به این سؤال که کدام‌یک از عناصر فرآیند توسعه محصول جدید، به بهره‌وری نوآوری منجر می‌شوند | شکل‌گیری ایده، غربالگری ایده، طرح پیشنهادی پروژه، تبیین ابعاد پروژه، شکل‌گیری مفهوم، ساخت و آزمایش نمونه اولیه، توسعه محصول، راه‌اندازی، بررسی‌های بعد از راه‌اندازی |

## ۳. روش‌شناسی پژوهش

تحقیق حاضر ازنظر هدف کاربردی و از نظر روش، توصیفی-پیمایشی است. سؤال اصلی تحقیق این است که الگوی فرآیندی توسعه محصولات نرم‌افزاری دارای چه ابعاد و روابطی است. برای پاسخگویی به این سؤال، از رویکرد مدل‌سازی ساختاری تفسیری (ISM) استفاده می‌شود. رویکرد ISM، به شناخت پیچیدگی روابط اجزای یک سیستم کمک می‌کند و یک قضاوت تفسیری درباره چگونگی روابط متغیرها ارائه می‌دهد. همچنین، این تکنیک مدل‌سازی،

1. Waterfall
2. Spiral model

ساختار کلی روابط میان متغیرها را درقالب یک نمودار نشان می‌دهد و متغیرها را درقالب سطوح مختلف طبقه‌بندی می‌کند [۳۲].

جامعه آماری تحقیق، متخصصان صنعت نرم‌افزار هستند که با توجه به مراحل مختلف پژوهش، در دو بخش طبقه‌بندی شده‌اند. بخشی از جامعه آماری شامل متخصصانی است که مستقیماً با فرآیند توسعه محصولات نرم‌افزاری درگیر هستند که یک بار در مرحله شناسایی ابعاد و شاخص‌های فرآیند و بار دیگر برای تحلیل مدل فرآیندی توسعه محصولات نرم‌افزاری، ازنظر آن‌ها استفاده شده است. محدودیت درنظرگرفته‌شده برای شناسایی این متخصصان، برخورداری از سابقه مشارکت در حداقل سه پروژه توسعه محصولات نرم‌افزاری بوده است. این محدودیت، اطمینان لازم برای اشراف جامعه آماری و نمونه‌ها را نسبت‌به فرآیند توسعه محصولات نرم‌افزاری ایجاد می‌کند. برای شناسایی متغیرهای مکنون تحقیق درقالب تحلیل عاملی، به بخش دوم جامعه آماری توجه شده است که شامل آن دسته از فعالان صنعت نرم‌افزار است که حداقل دارای مدرک کاردانی و دو سال سابقه حضور در این صنعت بوده‌اند. این بخش از جامعه آماری، شناختی نسبی از فرآیند توسعه و تولید محصولات نرم‌افزاری دارند و پاسخ‌های آن‌ها، برای استخراج عوامل اصلی و فرعی به‌کار رفته است. پایایی پرسش‌نامه براساس روش آلفای کرونباخ سنجیده شده است. عدد آلفای کرونباخ برای کل پرسش‌نامه، ۰/۸۰۵ به‌دست آمده است که با توجه به بالاتر بودن این عدد از سطح قابل‌قبول ۰/۷ [۳۰]، پایایی ابزار اندازه‌گیری قابل‌قبول ارزیابی می‌شود.

## ٤. تحلیل داده‌ها و یافته‌های پژوهش

برای دستیابی به مدل فرآیندی توسعه محصولات نرم‌افزاری براساس رویکرد ISM، گام‌های زیر برداشته شده است.

**شناسایی ابعاد و شاخص‌هایی که به مسئله مربوط هستند.** برای شناسایی شاخص‌های مرتبط با فرآیند توسعه محصولات نرم‌افزاری، ابتدا با بررسی ادبیات نظری، یک پرسش‌نامه بازپاسخ طراحی شد و در اختیار ۹ متخصص صنعت نرم‌افزار قرار گرفت. همان‌گونه که در توضیح جامعه آماری در قسمت روش‌شناسی بیان شده است، همه این افراد، سابقه مشارکت در حداقل سه پروژه توسعه محصولات نرم‌افزاری را داشتند. در این میان، چهار متخصص دارای تحصیلات کارشناسی و چهار نفر دارای کارشناسی ارشد بودند و یک نفر مدرک دکتری داشت. همچنین، به‌جز یک نفر، تحصیلات همه نمونه‌ها در حوزه رایانه بود. خروجی این مرحله، شناسایی ۲۲ متغیر (آیتم) در فرآیند توسعه محصولات نرم‌افزاری بوده است. تمامی این متغیرها ازجانب حداقل

پنج نفر از صاحب‌نظرانی که مورد سؤال قرار گرفتند، تأیید شدند. سپس، به‌منظور دسته‌بندی متغیرها و تعیین عامل‌های اصلی، پرسش‌نامه‌ای از متغیرهای شناسایی‌شده، درمیان نمونه‌ای متشکل از ۲۳۷ فعال صنعت نرم‌افزار توزیع شد. برای اطمینان از اشراف نمونه‌ها نسبت‌به موضوع تحقیق، محدودیت برخورداری از حداقل دو سال سابقه حضور در صنعت نرم‌افزار تعیین شد. همچنین، برای نمونه‌گیری، از روش دردسترس ساده استفاده شده است و همه نمونه‌ها ازمیان مدیران و کارکنان شرکت‌های حاضر در یک نمایشگاه تخصصی صنعت نرم‌افزار شناسایی شدند. تعداد نمونه‌ها از دو جهت، به افزایش اعتبار تحقیق کمک کرده است؛ نخست اینکه خطای سیستماتیک ناشی از استفاده از روش غیراحتمالی نمونه‌گیری را کاهش داده است [۳۴] و دوم اینکه، با توجه به استفاده از این داده‌ها برای تحلیل عاملی و لزوم وجود حداقل ۱۰ آزمودنی برای هر متغیر در این روش [۲۹]، کفایت اولیه نمونه‌ها فراهم شده است. ازنظر جمعیت‌شناختی، بخش قابل توجهی از نمونه‌های تحقیق مرد هستند و زنان تنها ۲۶ درصد از نمونه‌ها را به خود اختصاص داده‌اند. ازلحاظ تحصیلات، مطابق با داده‌های حاصل از پرسش‌نامه‌ها، بیشترین تعداد پاسخ‌دهندگان دارای مدرک کارشناسی (۵۱/۵ درصد) هستند و پس از آن، گروه کاردانی، کارشناسی ارشد و دکتری، به‌ترتیب بیشترین نمونه‌ها را با ٪۲۳/۸، ٪۲۰/۱ و ٪۴/۶ به خود اختصاص می‌دهند. میانگین سابقه کار نمونه‌ها ۱۲/۸ سال و میانگین سابقه مرتبط با صنعت نرم‌افزار، ۷/۷ سال بوده است.

پس از انجام تحلیل عاملی مرتبه اول، مشخص شد که تنها متغیر امکان‌سنجی مالی برای تجهیز تولید، همبستگی قابل توجهی با هیچ‌یک از فاکتورهای استخراج‌شده ایجاد نمی‌کند؛ بنابراین، تحلیل عاملی مرتبه دوم پس از حذف این متغیر انجام شد.

مقدار KMO به‌دست‌آمده برای هریک از مؤلفه‌های اصلی به‌کاررفته در مدل مفهومی تحقیق باید بزرگ‌تر از ۰/۶ و سطح معناداری کای دو حاصل از اجرای آزمون بارتلت برای مؤلفه‌های یادشده باید کمتر از ۰/۰۵ باشد. برای این تحقیق (مطابق جدول ۲)، شاخص KMO مقدار ۰/۷۸۳ به‌دست آمد. نتایج آزمون بارتلت نشان می‌دهد که سطح معناداری این آزمون کمتر از ۵ درصد است که بیانگر مناسب بودن ساختار مدل تحلیل عاملی است و فرض شناخته بودن ماتریس همبستگی رد می‌شود.

جدول ۲. نتایج آزمون بارتلت و شاخص KMO

| | |
|---|---|
| شاخص KMO | ۰/۷۸۳ |
| مقدار کای دو آزمون بارتلت | ۱۸۲/۴۲ |
| سطح معناداری آزمون بارتلت | ۰/۰۰۰ |

نتایج مربوط به تحلیل عاملی مرتبه دوم بیانگر شناسایی ۸ عامل اصلی است که با توجه به ادبیات نظری، نام‌گذاری می‌شوند. در جدول ۳، نام عامل‌های به‌دست‌آمده به‌ترتیب اهمیت در واریانس تبیین‌شده با استفاده از روش واریماکس چرخش‌یافته آمده است که درمجموع، این ۸ عامل، ۸۱/۴ درصد از کل واریانس فرآیند توسعه محصولات نرم‌افزاری را تبیین می‌کنند.

*جدول ۳. روش تحلیل مؤلفه اصلی و روش چرخش واریماکس با نرمال‌سازی*

| عوامل فرآیندی اصلی | درصد واریانس | درصد واریانس تجمعی |
|---|---|---|
| بازبینی تولید | ۱۸/۲ | ۱۸/۲ |
| ایده‌پردازی | ۱۵/۹ | ۳۴/۱ |
| تحلیل فرصت‌های بازار | ۱۲/۴ | ۴۶/۵ |
| توسعه نسخه‌های اصلاحی | ۱۱/۵ | ۵۷/۰ |
| توسعه طراحی | ۸/۲ | ۶۵/۲ |
| ایجاد مفهوم | ۶/۴ | ۷۱/۶ |
| مهندسی نمونه اولیه | ۵/۸ | ۷۷/۴ |
| تولید انبوه | ۳/۰ | ۸۱/۴ |

در ادامه‌ی تحلیل عاملی داده‌های تحقیق، براساس بارهای عاملی، ساختارهای هر عامل متشکل از متغیرهای مربوط به آن‌ها شناسایی شد و تنها عواملی در ساختار گنجانده شدند که دارای بار عاملی بیش از ۰/۵ هستند. ماتریس چرخش‌یافته متغیرهای تحقیق و ساختار عوامل جابه‌جاشده واریماکس شاخص‌ها در جدول ۴ آمده است.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی

*جدول ۴. ساختار عوامل جابه‌جاشده واریماکس شاخص‌ها*

| عوامل | بار عامل |
|---|---|
| عامل اول- تحلیل فرصت‌های بازار | |
| پویش نیازهای مشتریان | ۰/۷۵۱ |
| تحلیل روندهای بازار | ۰/۷۸۰ |
| عامل دوم- ایده‌پردازی | |
| ارائه ایده‌های نو | ۰/۶۱۸ |
| غربالگری ایده‌ها | ۰/۸۱۴ |
| ارزیابی ایده منتخب | ۰/۸۳۳ |
| عامل سوم- ایجاد مفهوم | |
| طراحی مفهومی | ۰/۵۹۷ |
| طراحی صنعتی | ۰/۶۲۱ |
| ارزیابی صنعتی | ۰/۸۰۴ |
| عامل چهارم- توسعه طراحی | |
| معماری سخت‌افزاری | ۰/۸۰۱ |
| معماری نرم‌افزاری | ۰/۶۵۳ |
| طراحی اجزای نرم‌افزار | ۰/۶۸۷ |
| طراحی نرم‌افزار یکپارچه | ۰/۷۷۰ |
| عامل پنجم- مهندسی نمونه‌ای | |
| کدنویسی اجزای نرم‌افزار | ۰/۸۲۴ |
| آزمودن اجزای نرم‌افزار | ۰/۷۳۵ |
| ساخت نرم‌افزار یکپارچه اولیه | ۰/۷۶۳ |
| آزمودن نرم‌افزار یکپارچه اولیه | ۰/۵۶۱ |
| عامل ششم- بازبینی تولید | |
| تعیین معیارهای ارزیابی | ۰/۸۱۲ |
| اجرای آزمایشی برنامه | ۰/۸۶۶ |
| عامل هفتم- تولید انبوه | |
| تجهیز تولید | ۰/۶۹۳ |
| تولید انبوه | ۰/۷۲۷ |
| عامل هشتم- توسعه نسخه‌های اصلاحی | |
| ارزیابی مشکلات کاربران | ۰/۵۱۵ |
| رفع ناسازگاری‌های محصول | ۰/۵۷۲ |
| ارائه نسخه‌های اصلاحی | ۰/۶۹۵ |

**ماتریس خودتعاملی ساختاری.** برای تحلیل مدل فرآیندی توسعه محصولات نرم‌افزاری، پرسش‌نامه‌ای تهیه شد که روابط دوبه‌دوی ابعاد شناسایی‌شده در مرحله قبل را می‌سنجد. این

پرسش‌نامه در اختیار ۳۵ متخصص قرار گرفت که در زمینه توسعه محصولات نرم‌افزاری دخیل بودند. همه این افراد، حداقل در سه پروژه توسعه محصولات نرم‌افزاری مشارکت داشتند و حداقل دارای مدرک کارشناسی در زمینه‌های مرتبط با رایانه و یا رشته‌های مهندسی بودند. از این میان، ۲۸ پرسش‌نامه برگشت داده شد که درمیان آن‌ها، تمامی ۹ متخصصی که در مرحله شناسایی ابعاد و شاخص‌ها همکاری کرده بودند نیز حضور داشتند. درمجموع، ازمیان این ۲۸ مشارکت‌کننده، ۱۶ نفر دارای تحصیلات کارشناسی، ۱۰ نفر دارای تحصیلات کارشناسی ارشد و و دو نفر دارای مدرک دکتری بودند. ۲۱ مشارکت‌کننده در رشته‌های مرتبط با رایانه و مابقی در رشته‌های مهندسی تحصیل کرده‌اند. نتایج حاصل از داده‌های گردآوری‌شده در این مرحله، در جدول ۵ ارائه شده است. برای ترسیم ماتریس خودتعاملی ساختاری، از چهار نشانه زیر استفاده شده است:

V: عنصر i به تحقق عنصر j کمک می‌کند؛

A: عنصر j به تحقق عنصر i کمک می‌کند؛

X: عناصر i و j هم‌زمان به تحقق یکدیگر کمک می‌کنند؛

O: عناصر i و j ارتباطی با هم ندارند.

*جدول ۵.ماتریس خودتعاملی ساختاری*

| ابعاد | ۱ | ۲ | ۳ | ٤ | ٥ | ٦ | ۷ | ۸ |
|---|---|---|---|---|---|---|---|---|
| | | X | O | O | O | O | O | O |
| | | | V | O | O | O | O | O |
| | | | | X | O | O | O | O |
| | | | | | X | O | O | O |
| | | | | | | X | O | O |
| | | | | | | | X | O |
| | | | | | | | | X |

**ماتریس دستیابی.** ماتریس ساختاری خودتعاملی برای اینکه روابط میان ابعاد و شاخص‌ها را تبیین کند، باید به یک ماتریس دوتایی تبدیل شود که ماتریس اولیه دستیابی نامیده می‌شود. این کار، از طریق جایگزین کردن اعداد ۰ و ۱ با نشانگرهای A، V، X و O صورت می‌گیرد [۴]. این جایگزینی براساس سازوکارهای زیر انجام شده است:

اگر ورودی (i, j) در ماتریس ساختاری خودتعاملی V است، در ماتریس دستیابی به (i, j)، عدد ۱ و به (j, i) عدد ۰ اختصاص می‌یابد.

اگر ورودی (i, j) در ماتریس ساختاری خودتعاملی A است، در ماتریس دستیابی به (i, j)، عدد ۰ و به (j, i) عدد ۱ اختصاص می‌یابد.

اگر ورودی (i, j) در ماتریس ساختاری خودتعاملی X است، در ماتریس دستیابی به (i, j)، عدد ۱ و به (j, i) عدد ۱ اختصاص می‌یابد.

اگر ورودی (i, j) در ماتریس ساختاری خودتعاملی O است، در ماتریس دستیابی به (i, j)، عدد ۰ و به (j, i) عدد ۰ اختصاص می‌یابد.

ماتریس دستیابی اولیه تنها روابط مستقیم میان عناصر را نشان می‌دهد؛ درحالی که یکی از مفروضات بنیادین ISM این است که اگر A به B و B به C مربوط است، پس A نیز به C مربوط است. برای اینکه ماتریس دستیابی اولیه، این روابط غیرمستقیم را نیز نشان دهد، باید آن را به توان K+1 رساند که به ماتریس حاصل، ماتریس دستیابی نهایی گفته می‌شود. در این ماتریس، برخی عناصر صفر تبدیل به ۱ خواهند شد که به‌صورت (*۱) نشان داده می‌شود. نتایج حاصل از ماتریس دستیابی اولیه و ماتریس دستیابی نهایی در جدول ۶ و ۷ آمده است.

*جدول ۶. ماتریس دستیابی اولیه*

| ابعاد | ۱ | ۲ | ۳ | ۴ | ۵ | ۶ | ۷ | ۸ |
|---|---|---|---|---|---|---|---|---|
| **۱** | ۱ | ۱ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| **۲** | ۱ | ۱ | ۱ | ۰ | ۰ | ۰ | ۰ | ۰ |
| **۳** | ۰ | ۰ | ۱ | ۱ | ۰ | ۰ | ۰ | ۰ |
| **٤** | ۰ | ۰ | ۱ | ۱ | ۱ | ۰ | ۰ | ۰ |
| **٥** | ۰ | ۰ | ۰ | ۱ | ۱ | ۱ | ۰ | ۰ |
| **٦** | ۰ | ۰ | ۰ | ۰ | ۱ | ۱ | ۱ | ۰ |
| **۷** | ۰ | ۰ | ۰ | ۰ | ۰ | ۱ | ۱ | ۱ |
| **۸** | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۱ | ۱ |

*جدول ۷. ماتریس دستیابی نهایی*

| ابعاد | ۱ | ۲ | ۳ | ٤ | ٥ | ٦ | ۷ | ۸ | قدرت نفوذ |
|---|---|---|---|---|---|---|---|---|---|
| **۱** | ۱ | ۱ | ۱* | ۰ | ۰ | ۰ | ۰ | ۰ | ۳ |
| **۲** | ۱ | ۱ | ۱ | ۱* | ۰ | ۰ | ۰ | ۰ | ۴ |
| **۳** | ۰ | ۰ | ۱ | ۱ | ۱* | ۰ | ۰ | ۰ | ۳ |
| **٤** | ۰ | ۰ | ۱ | ۱ | ۱ | ۱* | ۰ | ۰ | ۴ |
| **٥** | ۰ | ۰ | ۰ | ۱ | ۱ | ۱ | ۱* | ۰ | ۴ |
| **٦** | ۰ | ۰ | ۰ | ۰ | ۱ | ۱ | ۱ | ۱* | ۴ |
| **۷** | ۰ | ۰ | ۰ | ۰ | ۰ | ۱ | ۱ | ۱ | ۳ |
| **۸** | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۱ | ۱ | ۲ |
| **میزان وابستگی** | ۲ | ۲ | ۴ | ۴ | ۴ | ۴ | ۴ | ۳ | |

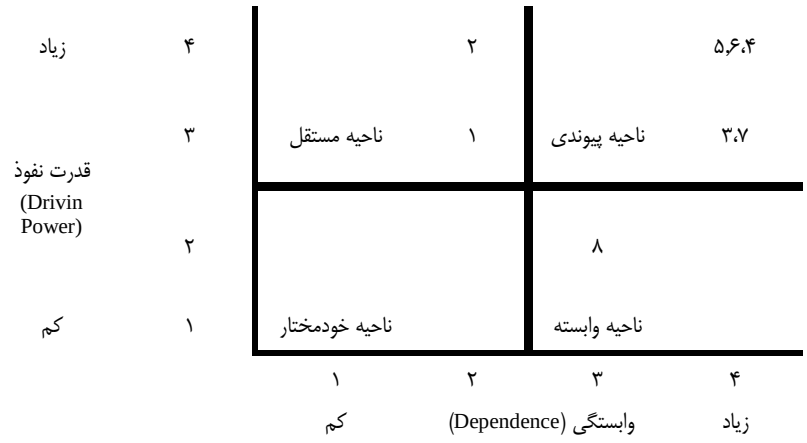
**تعیین روابط و سطوح.** از ماتریس دستیابی، مجموعه عوامل دستیابی مقدم و مشترک برای هر عنصر قابل شناسایی است. مجموعه دستیابی شامل عواملی است که سایر عوامل به تحقق آن‌ها کمک می‌کنند؛ درحالی که مجموعه مقدم به تحقق سایر عوامل کمک می‌کنند [۱۶]. براساس ماتریس دستیابی، یکی از عناصر که در بالاترین سطح سلسله‌مراتب ISM قرار می‌گیرد و هیچ‌یک از عوامل دیگر بر آن اثر نمی‌گذارد، مشخص می‌شود. درادامه، عنصری که در مرتبه بعدی بیشترین تأثیر را از سایر عوامل می‌پذیرد، تعیین می‌شود و این فرآیند تا تعیین سطح تمامی عناصر ادامه می‌یابد. تعیین این سطوح به ترسیم نمودار و مدل نهایی مبتنی‌بر ISM کمک می‌کند.

*جدول ۸. روابط و سطوح ابعاد فرآیند توسعه محصولات نرم‌افزاری*

| ابعاد | مجموعه خروجی | مجموعه ورودی | مجموعه مشترک | سطح |
|---|---|---|---|---|
| ۱ | ۱،۲،۳ | ۱،۲ | ۱،۲ | III |
| ۲ | ۱،۲،۳،۴ | ۱،۲ | ۱،۲ | III |
| ۳ | ۳،۴،۵ | ۱،۲،۳،۴ | ۳،۴ | III |
| ۴ | ۳،۴،۵،۶ | ۲،۳،۴،۵ | ۳،۴،۵ | II |
| ۵ | ۴،۵،۶،۷ | ۳،۴،۵،۶ | ۴،۵،۶ | II |
| ۶ | ۵،۶،۷،۸ | ۴،۵،۶،۷ | ۵،۶،۷ | II |
| ۷ | ۶،۷،۸ | ۵،۶،۷،۸ | ۶،۷،۸ | I |
| ۸ | ۷،۸ | ۶،۷،۸ | ۷،۸ | I |

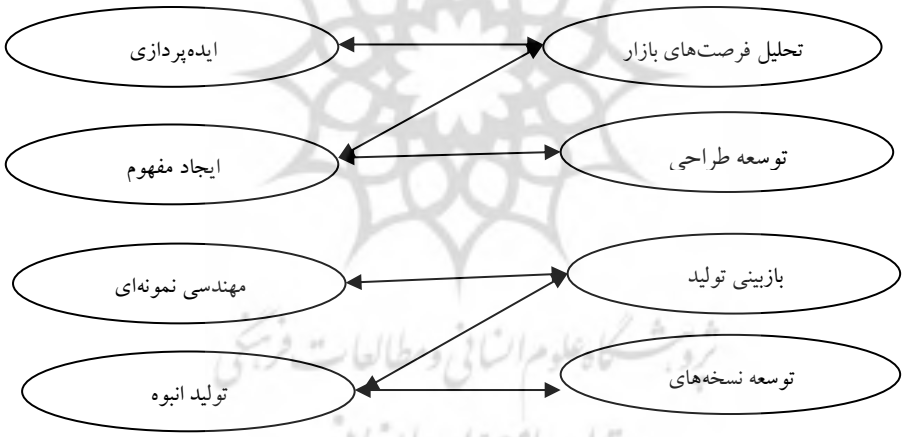
یکی از مزایای رویکرد ISM، امکان تقسیم‌بندی عناصر است [۳۲] که در این تحقیق، از رویکرد مندال و دشماخ (۱۹۹۴) استفاده می‌شود [۲۵]. در این رویکرد، همه عناصر براساس قدرت نفوذ و قدرت وابستگی که در ماتریس دستیابی نهایی (جدول ۷) محاسبه شده بود، در چهار طبقه، شامل عناصر مستقل، وابسته، پیوندی و خودمختار، قرار می‌گیرند.

خوشه‌بندی ابعاد برمبنای قدرت محرک و قدرت وابسته در شکل ۱ نشان داده شده است. همان‌گونه که این شکل نشان می‌دهد، هیچ عنصری در ناحیه خودمختار که قدرت محرک و قدرت وابستگی ضعیفی دارد و تقریباً از سیستم منقطع است، قرار نگرفته است. در ناحیه وابسته که قدرت محرک ضعیف و قدرت وابستگی قوی دارند، تنها عنصر ۸ قرار گرفته است. عناصری که در ناحیه سوم (ناحیه پیوندی) قرار می‌گیرند، قدرت محرک و وابستگی قوی دارند. ازمیان عناصر اصلی توسعه محصولات نرم‌افزاری در این تحقیق، عناصر ۳، ۴، ۵، ۶ و ۷ در ناحیه پیوندی قرار می‌گیرند. خوشه چهارم، شامل عناصر مستقل است که قدرت محرک قوی و قدرت وابستگی ضعیف دارند. عناصر ۱ و ۲ در این ناحیه قرار می‌گیرند.




*شکل ۱. خوشه‌بندی ابعاد براساس قدرت نفوذ- وابستگی*

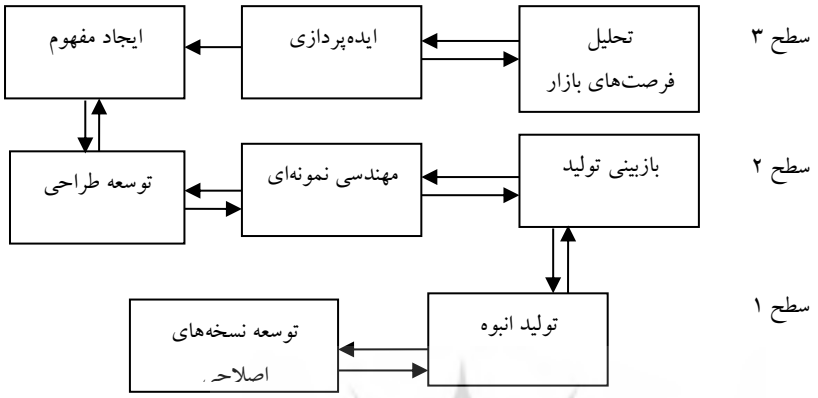
**ترسیم نمودار شبکه تعاملات ابعاد.** مدل ساختاری از ماتریس دستیابی اولیه استخراج می‌شود. اگر بین عناصر i و j رابطه وجود داشته باشد، ازطریق ترسیم خطی از i به j نشان داده می‌شود. این نمودار، نمودار ساختاری اولیه یا دیاگراف نامیده می‌شود.




*شکل ۲. نمودار ساختاری اولیه توسعه محصولات نرم‌افزاری*

پس از حذف انتقالات، نمودار نهایی شکل می‌گیرد. این نمودار که مدل مبتنی‌بر ISM نام دارد، روابط و سطوح ابعاد فرآیندی توسعه محصولات نرم‌افزاری را مطابق مرحله چهارم روش تحقیق نشان می‌دهد. مطابق شکل ۳، مدل فرآیندی در سه سطح قابل طبقه‌بندی است که در

سطح نخست که شالوده مدل است، تولید انبوه و توسعه نسخه‌های اصلاحی قرار دارد. به‌دلیل فرآیندی بودن مدل، سطح نخست به‌معنای مراحل نهایی فرآیند است. سطح دوم شامل سه گام فرعی توسعه طراحی، مهندسی نمونه‌ای و بازبینی تولید است و در آن، مفهوم شکل‌گرفته آماده تولید انبوه می‌شود. در سطح سوم، گام‌های ابتدایی فرآیند توسعه محصولات نرم‌افزاری قرار دارد که شامل تحلیل فرصت‌های بازار، ایده‌پردازی و ایجاد مفهوم است.




*شکل ۳. مدل فرآیندی توسعه محصولات نرم‌افزاری براساس روش ISM*

## ۵. نتیجه‌گیری و پیشنهادها

در مطالعه حاضر، براساس رویکرد مدل‌سازی ساختاری تفسیری (ISM)، یک الگوی فرآیندی برای توسعه محصولات نرم‌افزاری ارائه شد. براساس این الگو، برای توسعه محصولات نرم‌افزاری جدید، باید ۸ گام اصلی و ۲۳ گام فرعی برداشته شود.

گام نخست توسعه محصولات نرم‌افزاری، تحلیل فرصت‌های بازار است. در بیشتر مطالعات پیشین، به تحلیل فرصت‌ها به‌عنوان یکی از عوامل مؤثر بر موفقیت فرآیند توسعه محصول دیده شده بود، نه بخشی از این فرآیند [۲۴]. با این حال، بازار نرم‌افزارها از چنان تلاطم و پیچیدگی برخوردار است که فرصت‌محور بودن ایده اولیه جزئی از فرآیند توسعه محصول تلقی می‌شود [۲۶]. براساس این گام مقدماتی، برای توسعه محصولات، ابتدا باید بازارشناسی دقیقی صورت گیرد و نیازهای مشتریان و روندهای بازار ارزیابی و تحلیل شوند.

در گام دوم، بر اساس تحلیل فرصت‌ها در گام پیشین، ایده‌های اولیه ارائه و ارزیابی می‌شوند. با توجه به امکان وجود ایده‌های متنوع و محدودیت‌های شرکت‌های نرم‌افزاری برای سرمایه‌گذاری روی ایده‌های جدید، غربالگری ایده‌ها نقش مهمی در توسعه محصول دارد. برای

غربالگری ایده‌ها می‌توان از روش‌های مختلفی استفاده کرد؛ مانند طبقه‌بندی سه‌گانه ABC، رویکرد تحلیل سلسله‌مراتبی AHP، رای‌گیری بی‌نام و چک‌لیست‌های بسیار متنوعی مانند پرسش‌نامه مرکز تحقیقات خلاقیت پرینستون[1] و چک‌لیست توبیا (۲۰۰۷) که مبتنی‌بر دیدگاه مشتریان است [۳۵].

در گام سوم پس از انتخاب ایده نهایی، مفهوم محصول ایجاد می‌شود که به‌معنی تعیین مزایا، اجزا و ویژگی‌های کلیدی محصول برای مشتریان است و ازطریق آن می‌توان، فرآیند توسعه محصول را منسجم، هدفمند و شفاف کرد. گام ایجاد مفهوم، شامل دو گام فرعی طراحی مفهومی و ارزیابی مفهوم است که در طراحی مفهومی، یک شمای کلی از اجزا و ابعاد نرم‌افزار مطابق با ایده ارائه‌شده ترسیم و مشخص می‌شود و در گام ارزیابی، امکان‌سنجی فنی و مالی هریک از این اجزا و ابعاد صورت می‌گیرد. باید توجه داشت که هرچه مفهوم شفاف‌تر و دقیق‌تر تدوین شود، گام‌های مرتبط با طراحی و ساخت پیچیدگی و دشواری کمتری خواهند داشت.

گام چهارم توسعه محصولات نرم‌افزاری، توسعه فرآیند طراحی است که درمورد این محصولات، تفاوت قابل توجهی با محصولات سخت‌افزاری دارد. یکی از این تفاوت‌ها لزوم توجه به معماری نرم‌افزاری و سخت‌افزاری محصول است. در این معماری، ساختارهای کلی سیستم، اجزای سیستم و روابط بین اجزا، هم در بعد نرم‌افزاری و هم در بعد سخت‌افزاری، تعیین می‌شود. اهمیت معماری به لزوم سازگاری اجزای محصولات نرم‌افزاری بازمی‌گردد؛ زیرا اگر اجزای یک سیستم نرم‌افزاری با هم سازگار نباشند، عملکرد جزئی و حتی کلی سیستم با اشکال مواجه خواهد شد. پس از اطمینان درمورد صحت و سازگاری معماری محصول، همه اجزا درقالب یک سیستم یکپارچه طراحی می‌شوند.

پس از تکمیل گام توسعه طراحی، نمونه اولیه از نرم‌افزار در چهار گام فرعی ساخته می‌شود. دو گام فرعی به فعالیت‌های اجرایی و دو گام دیگر، به فعالیت‌های کنترلی مربوط است. در مهندسی نمونه اولیه نیز مانند گام طراحی، ابتدا اجزای نرم‌افزار آماده می‌شوند و سپس یکپارچه‌سازی صورت می‌گیرد. در پایان هر مرحله، برای اطمینان از صحت عملکرد اجزا و سیستم یکپارچه، آزمونی صورت می‌گیرد. این آزمون هم شامل مقایسه کدهای نوشته‌شده با طراحی مفهومی و طراحی نهایی است و هم شامل بررسی عملکرد واقعی نرم‌افزار.

مطابق گام‌های استخراج‌شده با رویکرد ISM، پیش از تولید انبوه نرم‌افزار، باید بازبینی نهایی در محصول صورت گیرد. در این گام، ابتدا معیارهای ارزیابی موفقیت محصول در بازار مشخص می‌شود و سپس براساس این معیارها، یک آزمایش مقدماتی روی یک نمونه یا پنل از بازار

1. Princeto nCreative Research

صورت می‌گیرد تا اگر محصول فاقد ویژگی‌های مطلوب تشخیص داده شد، برای اصلاحات لازم به مراحل قبل و به‌ویژه توسعه طراحی بازگردد.

پس از اطمینان از واکنش مناسب و مطلوب بازار نسبت به عرضه محصولات نرم‌افزاری، در گام هفتم، اقدام به تولید انبوه می‌شود. نخستین گام فرعی در تولید انبوه تجهیز تولید است که به‌معنی پیش‌بینی و فراهم کردن تمامی مقدمات و امکانات لازم برای تولید است. پس از اطمینان از تجهیز تولید، محصولات در حجم انبوه تولید و عرضه می‌شوند.

گام پایانی توسعه محصولات نرم‌افزاری، توسعه نسخه‌های اصلاحی است. افزوده شدن این گام به مدل فرآیندی تحقیق حاضر که در مطالعات پیشین کمتر مورد توجه قرار گرفته بود، بیانگر این است که تولید انبوه پایان خط توسعه محصولات نرم‌افزاری نیست و توسعه‌دهندگان محصول، مسئول اخذ بازخور از بازارهای هدف و مصرف‌کنندگان و رفع مشکلات احتمالی، درقالب ارائه نسخه‌های اصلاحی به بازار هستند. نکته قابل‌توجه این است که در گام بازبینی تولید نیز وضعیت بازار و احتمال موفقیت محصول ارزیابی می‌شود؛ اما تا محصول به‌صورت انبوه به بازار عرضه نشود، برخی مشکلات مشخص نخواهند شد.

در یک جمع‌بندی نهایی از یافته‌های تحقیق، چند نکته در تفاوت توسعه محصولات نرم‌افزاری با محصولات سخت‌افزاری قابل استخراج است. نکته نخست این است که در توسعه محصولات نرم‌افزاری، باید بر ایده‌هایی تمرکز کرد که درمورد آن‌ها، تحلیل بازار صورت گرفته باشد. البته این به این معنی نیست که برای ایده‌پردازی درمورد توسعه محصولات سخت‌افزاری، تحلیل بازار صورت نمی‌گیرد؛ بلکه به‌معنی درجه اهمیت تحلیل بازار تا حدی است که به‌عنوان گام ورودی ایده‌پردازی برای محصولات نرم‌افزاری مطرح می‌شود. نکته دوم در مدل استخراجی، ترکیب سه گام سطح ۲ مدل است که از توسعه طراحی آغاز می‌شود و پس از مهندسی نمونه اولیه و پیش از تولید انبوه، بازبینی تولید صورت می‌گیرد. در سایر مدل‌های توسعه محصول که بیشتر با محصولات سخت‌افزاری متناسب هستند، دو گام توسعه و آزمودن نمونه اولیه و توسعه محصول مدنظر قرار گرفته است [۸و۳۶] که تفاوت اصلی آن با گام‌های مدل تحقیق حاضر در بازبینی تولید است؛ به‌عبارت دیگر، پس از آماده شدن نمونه اولیه، باید شرایط عرضه عمومی محصول را با تعیین معیارهای دقیق ارزیابی کرد. نکته نهایی که در مدل‌های پیشین کمتر به آن توجه شده بود، توسعه نسخه‌های اصلاحی محصول به‌عنوان یک گام کلیدی در توسعه محصول است. در برخی مدل‌ها، ارزیابی‌های پس از راه‌اندازی یا تولید انبوه به‌عنوان آخرین گام توسعه محصول مطرح شده است [۱۷] که با یک حلقه بازخور به ابتدای فرآیند توسعه محصول بازمی‌گردد؛ اما درمورد محصولات نرم‌افزاری، تنها ارزیابی کافی نیست و نسخه‌های اصلاحی به‌عنوان بخشی از فرآیند توسعه محصول فعلی دیده می‌شوند.

در راستای نتایج تحقیق حاضر، پیشنهادهای زیر برای توسعه محصولات نرم‌افزاری ارائه می‌شود:

۱. توسعه‌دهندگان محصولات نرم‌افزاری لازم است در نخستین گام فرآیند توسعه، درک مناسبی از بازار به‌دست آورند و نیازها و فرصت‌های موجود را به‌خوبی تحلیل کنند. این بازارشناسی، احتمال موفقیت محصول نهایی را در بازار افزایش می‌دهد و از مخاطرات و اقدامات پرهزینه ناشی از توسعه محصولات ناموفق می‌کاهد.

۲. در طراحی محصولات نرم‌افزاری، باید نگرش سیستمی حاکم باشد تا از بهینه‌سازی بخشی جلوگیری شود و هزینه‌های ناشی از ناسازگاری‌ها به حداقل برسد. در این راستا، به توسعه‌دهندگان محصولات نرم‌افزاری توصیه می‌شود که در طراحی و معماری نرم‌افزار، سازگاری اجزا با عملکرد کلی نرم‌افزار و سخت‌افزار و سازگاری نرم‌افزارهای درحال توسعه با استانداردهای سخت‌افزاری را در نظر بگیرند.

۳. بیشتر گام‌های توسعه محصول دوطرفه است و این به‌معنی امکان بازگشت به گام قبل و گام‌های قبل‌تر، برای انجام اقدام اصلاحی است. از آنجا که هرچه اقدامات اصلاحی در گام‌های نخست صورت گیرد، هزینه‌های کمتری به توسعه‌دهندگان تحمیل می‌شود، توصیه می‌شود که هر گام، علاوه‌بر اینکه مقدمه‌ای برای گام بعدی است، کنترلی بر گام پیشین نیز باشد؛ به‌ویژه در گام بازبینی تولید که حلقه واسط طراحی و اجرا است، براساس استانداردهای ازپیش‌تعیین‌شده، ارزیابی دقیقی صورت گیرد تا درصورت نیاز به اقدامات اصلاحی، از هزینه‌های سنگین تولید انبوه اجتناب شود.

۴. توسعه‌دهندگان نرم‌افزار همواره باید برای ارائه نسخه‌های اصلاحی آمادگی داشته باشند. بخشی از این آمادگی به پیش‌بینی هزینه‌های این نسخه‌ها مربوط است که معمولاً مشتریان حاضر نیستند آن‌ها را بپردازند و بخشی از سود کل پروژه توسعه را کاهش می‌دهد.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی

## منابع

۱. اسماعیل پور، رضا، و جباری، مریم. (۱۳۸۹). بررسی فرآیند توسعه محصول جدید از ایده یابی تا عرضه به بازار، *چهارمین کنفرانس ملی مدیریت تکنولوژی ایران*، تهران.

2. Adolph, S., Kruchten, P., & Hall, W. (2012). Reconciling perspectives: A grounded theory of how people manage the process of software development. *Journal of Systems and Software, 85(6),* 1269-1286.

3. Ambler, S.W. (2005). Quality in an agile world. *Software Quality Professional, 7(4),* 34-40.

4. Attri, R., Nikhil, D.N., & Sharma, V. (2013). Interpretive structural modeling (ISM) approach: an overview. *Research Journal of Management Sciences, 2(2),* 3-8.

5. Barclay, I., & Dann, Z. (2000). Management and Organizational Factors in New Product Development (NPD) Success, *Concurrent Engineering, 8 (2),* 115-132.

6. Barczak, G., & Kahn, K.B. (2012). Identifying new product development best practice, *Business Horizons, 55,* 293-305.

7. Boehm, B.W. (1988). A spiral model of software development and enhancement. *Computer, May,* 61-72.

8. Cho, S.H., & Eppinger, S.D. (2005). A Simulation-based Process Model for Managing Complex Design Projects, *IEEE Transactions on Engineering Management, 52(3),* 316-328.

9. Cockburn, A. (2001). *Agile Software Development.* 1st ed., Boston: Addison-Wesley.

10. Coenen, M., Kok, R.A.W. (2014). Workplace flexibility and new product development performance: The role of telework and flexible work schedules, *European Management Journal, 32(4),* 564-576.

11. Coleman, G., & O'Connor, R.V. (2008). An investigation into software development process formation in software start-ups. *Journal of Enterprise Information Management, 21(6),* 633-648.

12. Cooper, R.G. and Kleinschmidt, E.J. (1991). New product processes at leading industrial firms. *Industrial Marketing Management, 20(2),* 137-147.

13. Corallo, A., Laubacher, R., Margherita, A., & Turrisi, G. (2009). Enhancing product development through knowledge-based engineering (KBE): A case study in the aerospace industry. *Journal of Manufacturing Technology Management, 20 (8),* 598-614.

14. Durrani, U.K., Pita, Z., & Richardson, J. (2014). Coexistence of agile and SCM practices: An exploratory study of Australian agile software development organizations. *Journal of Information & Software Technology, 16(1),* 20-39.

15. Engwall, M., Kling, R., & Werr, A. (2005). Models in action: how management models are interpreted in new product development. *R & D Management, 35(4),* 427-439.

16. Govindan, K., Palaniappan, M., Zhu, Q., & Kannan, D. (2012). Analysis of third party reverse logistics provider using interpretive structural modeling. *International Journal of Production Economics, 140(1),* 204-211.

17. Grunert, K.G., & van Trijp, H.C.M. (2014). Consumer-Oriented New Product Development. *Encyclopedia of Agriculture and Food Systems, 2,* 375-386.

18. Harmancioglu, N., McNally, R.C., Calantone, R.J. & Durmusoglu, S.S. (2007). Your new product development (NPD) is only as good as your process: an

exploratory analysis of new NPD process design and implementation. *R&D Management 37*, 5, pp. 399-424.

19. Highsmith, J. (2004). Agile Project Management, Addison-Wesley, Boston, MA.

20 .Hoyer,W.D., Chandy, R., Dorotic, M., Krafft, M., and Singh, S.S. (2010). Consumer Cocreation in New Product Development, *Journal of Service Research, 13(3),* 283-296.

21. Krishnan, R.T., & Prabhu, G.R. (2002). Innovation in the Indian Information Technology Industry: A Study of the Software Product Development Process. Science, *Technology and Society, 7 (1),* 91-115.

22. Kettunen, J., Grushka-Cockayne, Y., Degraeve, Z., & De Reyck, B. (2015). New product development flexibility in a competitive environment. *European Journal of Operational Research, 244(3),* 892-904.

23. Luh, D-B., Ko, Y-T., & Ma, C-H. (2009). A Dynamic Planning Approach for New Product Development. *Concurrent Engineering, 17 (1),* 43-59.

24. Magdaleno, A.M., Werner, M, L., de Araujo, R.M. (2012). Reconciling software development models: A quasi-systematic review, *Journal of Systems and Software, 85(2),* 351-369.

25. Mandal, A. & Deshmukh, S. (1994). Vendor selection using interpretive structural modeling (ISM). *International Journal of Operations and Production Management, 14(6),* 52-59.

26.Martín, Y-l., & Yelmo, J.C. (2014). Guidance for the Development of Accessibility Evaluation Tools Following the Unified Software Development Process. *Procedia Computer Science, 27,* 302-311.

27. Misra, S., Kumar,V., Kumar, U., Fantazy, K., Akhter, M. (2012). Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management, 29(9),* 972-980.

28. Misra, S., & Singh, V. (2015). Conceptualizing open agile software development life cycle (OASDLC) model, *International Journal of Engineering & Technology, 32 (3),* 214-235.

29. Nunnally, J.M. (1994). *Psychometric Theory*. Third edition, New York: McGraw-Hill.

30. Osborne, J.W. (2008). *Best Practices in Quantitative Methods.* SAGE Publications, Inc.

31. Pitta, D., & Pitta, E. (2012). Transforming the nature and scope of new product development. *Journal of Product & Brand Management, 21(1),* 35-46.

32. Raj, T., Shankar, R., & Suhaib, M. (2008). An ISM approach for modeling the enablers of flexible manufacturing system: the case for India. *International Journal of Production Research, 46(24),* 6883-6912.

33. Royce, W. (1970). Managing the Development of Large Software Systems. *Institute of Electrical and Electronics engineers (IEEE), WESCON , 26,* 1-9.

34. Smith, T.F.M. (1983). On the validity of Inferences from Non-random Samples. *Journal of the Royal Statistical Society, 146,* 394-403.

35. Toubia, O. (2007). Adaptive Idea Screening Using Consumers. *Marketing Science, 26(3),* 342-360.

36. Tzokas, N., Hultink, E.J., & Hart, S. (2004). Navigating the new product development process. *Industrial Marketing Management, 33,* 619-626.