

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

مطالعه موردی و ارزیابی خط تولید نرم‌افزاری شرکت اعوان

ابوالفضل انصاری - ۹۰۲۰۹۸۴۴

محمد رضا بشارتی - ۹۰۳۰۰۴۳۹

موسی مرادی - ۹۰۲۱۰۵۸۳

شایان افراسیابی - ۹۰۲۱۰۳۵۶

تیر و مردادماه ۱۳۹۱

## فهرست مطالب

چکیده.....	۸
۱- تعریف اصطلاحات مرتبط.....	۹
۱-۱- نیاز به خط تولید.....	۹
۲-۱- مهندسی خط تولید نرم‌افزاری.....	۹
۳-۱- چارچوب نرم‌افزاری.....	۹
۴-۱- شخصی‌سازی گسترده.....	۹
۵-۱- نقاط تغییر و متغیرها.....	۱۰
۶-۱- چارچوب ارائه شده برای مهندسی خط تولید.....	۱۰
۷-۱- جمع‌بندی.....	۱۱
۲- معرفی شرکت اعوان.....	۱۲
۱-۲- مقدمه.....	۱۲
۲-۲- محورهای اصلی فعالیت.....	۱۲
۱-۲-۲- تولید نرم‌افزارهای سفارشی.....	۱۲
۲-۲-۲- خدمات آموزشی و مشاوره‌ای.....	۱۲
۳-۲-۲- تولید بسته‌های نرم‌افزاری.....	۱۲
۳-۲- مشتری‌ان عمده.....	۱۲
۴-۲- محصولات نرم‌افزاری.....	۱۳
۱-۴-۲- مدیریت ارتباط با شهروند.....	۱۳
۲-۴-۲- پورتال سازمانی.....	۱۴
۳-۴-۲- خط تولید نرم‌افزار (WiSE Solutions).....	۱۶
۴-۴-۲- سیستم مدیریت اسناد (DBFS).....	۱۶
۵-۴-۲- سیستم مدیریت گزارش‌های پویا.....	۱۷
۶-۴-۲- PetroViz.....	۱۸

- ۱۸-۷-۴-۲ ابزار نصب و به‌روزرسانی (WiSE Installer).....
- ۱۹-۸-۴-۲ ابزار پایگاه داده (WiSE DB Tools).....
- ۱۹-۹-۴-۲ جدول همراه.....
- ۱۹-۱۰-۴-۲ پرتال اینترنتی.....
- ۲۰-۵-۲ سایر خدمات شرکت اعوان.....
- ۲۰-۱-۵-۲ تولید نرم‌افزار سفارشی.....
- ۲۰-۲-۵-۲ پشتیبانی.....
- ۲۱-۳-۵-۲ مشاوره.....
- ۲۱-۴-۵-۲ نظارت.....
- ۲۱-۵-۵-۲ برون‌سپاری.....
- ۲۱-۶-۵-۲ آموزش.....
- ۲۳-۳ معرفی معماری خط تولید شرکت اعوان.....
- ۲۳-۱-۳ مقدمه.....
- ۲۳-۲-۳ معرفی wise.....
- ۲۴-۳-۳ امکانات چارچوب wise.....
- ۲۵-۳-۴ ساختار برنامه‌های توسعه داده شده با wise.....
- ۲۷-۵-۳ معرفی ارتباط مولفه‌های core و سایر برنامه‌ها.....
- ۳۰-۴ معرفی مدل BAPO و FEF.....
- ۳۰-۱-۴ مقدمه.....
- ۳۰-۱-۱-۴ تاریخچه FEF.....
- ۳۱-۲-۱-۴ ارتباط FEF با سایر رویکردها.....
- ۳۲-۳-۱-۴ ارزیابی خانواده محصولات نرم‌افزاری.....
- ۳۳-۴-۲-۴ مدل BAPO.....
- ۳۵-۴-۳-۴ چارچوب FEF.....

- ۳۶.....۴-۴- بعد کسب و کار (Business).....
- ۳۷.....Project based : سطح اول: ۴-۴-۱.....
- ۳۸.....Aware : سطح دوم: ۴-۴-۲.....
- ۳۹.....Managed : سطح سوم: ۴-۴-۳.....
- ۳۹.....Measured : سطح چهارم: ۴-۴-۴.....
- ۴۰.....Optimised : سطح پنجم: ۴-۴-۵.....
- ۴۱.....۴-۴-۶- خلاصه.....
- ۴۲.....۴-۵- بعد معماری.....
- ۴۳.....۴-۵-۱- سطح ۱: توسعه‌ی مستقل محصولات.....
- ۴۳.....۴-۵-۲- سطح ۲: زیرساخت مستقل از دامنه، ولی استاندارد شده.....
- ۴۴.....۴-۵-۳- سطح ۳: پلتفرم نرم‌افزاری.....
- ۴۴.....۴-۵-۴- سطح ۴: محصولات گوناگون قابل اشتقاق.....
- ۴۵.....۴-۵-۵- سطح ۵: اشتقاق خودکار محصولات.....
- ۴۵.....۴-۵-۶- خلاصه.....
- ۴۶.....۴-۶- بعد فرایند.....
- ۴۷.....۴-۶-۱- سطح اول: Initial.....
- ۴۷.....۴-۶-۲- سطح دوم: Managed.....
- ۴۸.....۴-۶-۳- سطح سوم: Defined.....
- ۵۰.....۴-۶-۴- سطح چهارم: Quantitatively managed.....
- ۵۱.....۴-۶-۵- سطح پنجم: Optimising.....
- ۵۱.....۴-۶-۶- خلاصه.....
- ۵۲.....۴-۷- بعد سازمان.....
- ۵۳.....۴-۷-۱- سطح اول: Project.....
- ۵۳.....۴-۷-۲- سطح دوم: Re-use.....

۵۴	.....Weakly connected سوم: سطح ۳-۷-۴
۵۴	.....Synchronised چهارم: سطح ۴-۷-۴
۵۵	.....Domain engineering پنجم: سطح ۵-۷-۴
۵۵	.....خلاصه ۶-۷-۴
۵۶	.....اجرای ارزیابی ۸-۴
۵۶	.....برای برآورد وضعیت FEF-۱-۸-۴
۵۷	.....به عنوان محک FEF-4-8-2
۵۸	.....برای ارتقا FEF-۳-۸-۴
۵۸	.....ارتباط با دیگر رویکردها ۴-۸-۴
۵۹	.....سازمان‌های پیچیده ۹-۴
۶۰	.....معماری ساخت یافته ۱۰-۴
۶۱	.....پرسشنامه ارزیابی خط تولید بر اساس مدل FEF ۵-۴
۶۱	.....مقدمه ۱-۵
۶۱	.....ربع کسب و کار (Business) ۲-۵
۶۲	.....ربع معماری (Architecture) ۳-۵
۶۲	.....سطح ۱ ۱-۳-۵
۶۲	.....سطح ۲ ۲-۳-۵
۶۲	.....سطح ۳ ۳-۳-۵
۶۳	.....سطح ۴ ۴-۳-۵
۶۳	.....سطح ۵ ۵-۳-۵
۶۳	.....ربع فرآیند (Process) ۴-۵
۶۳	.....5-4-1- Domain Engineering
۶۵	.....5-4-2- Application Engineering
۶۶	.....5-4-3- Collaboration

- ۵-۵-۵- ربع سازمان..... ۶۸
- ۵-۵-۱- سطح اول..... ۶۸
- ۵-۵-۲- سطح دوم..... ۶۸
- ۵-۵-۳- سطح سوم..... ۶۸
- ۵-۵-۴- سطح چهارم..... ۶۸
- ۵-۵-۵- سطح پنجم..... ۶۹
- 6- نتایج ارزیابی خط تولید شرکت اعوان با استفاده از مدل FEF..... ۷۰
- ۶-۱- مقدمه..... ۷۰
- ۶-۲- موارد کلی پیرامون خط تولید شرکت اعوان..... ۷۰
- 6-3- ارزیابی ربع کسب و کار..... ۷۶
- ۶-۴- ارزیابی ربع معماری..... ۷۷
- ۶-۵- ارزیابی ربع فرآیند..... ۷۸
- ۶-۶- ارزیابی ربع سازمان..... ۹۹
- ۷- بررسی مقالات مرتبط با رویکرد اعوان در زمینه خط تولید..... ۱۰۰
- ۷-۱- مقدمه..... ۱۰۰
- ۷-۲- تغییرپذیری یا ترکیب؟..... ۱۰۰
- ۷-۳- تأثیر محدوده بر خط تولید..... ۱۰۱
- 7-4- لزوم حفظ تعادل بین دو رویکرد بالابه پایین و پایین به بالا..... ۱۰۲
- ۷-۵- ترکیب دقیقاً به چه معناست؟..... ۱۰۴
- 7-6- جمع بندی و مقایسه..... ۱۰۴
- ۷-۷- یک مثال: SPLI..... ۱۰۶
- ۸- توصیه ها به شرکت اعوان..... ۱۱۰
- ۸-۱- مقدمه..... ۱۱۰
- ۸-۲- ربع کسب و کار..... ۱۱۰

۱۱۰.....۳-۸- ربع معماری

۱۱۰.....۴-۸- ربع فرآیند

۱۱۰.....۵-۸- ربع سازمان

۱۱۱.....مراجع

در این مقاله قصد داریم به بررسی یک مطالعه موردی<sup>1</sup> ویژه از به کارگیری فناوری خط تولید برای ایجاد یک خانواده از محصولات نرم‌افزاری پردازیم. برای این مطالعه، شرکت «مشاوران نرم‌افزاری/اعوان» انتخاب شده است که از جمله شرکت‌های پیشتاز، با سابقه و خوش نام در عرصه خط تولید محصولات نرم‌افزاری می‌باشد. بدین ترتیب پس از مطالعه مقالات مرتبط در زمینه ارزیابی خط تولید و آماده‌سازی یک پرسش‌نامه مدون، جلسات متعددی میان نگارندگان این مقاله و مسئولان شرکت اعوان برگزار شد که بر این اساس مقاله جاری تهیه شده است. بدون شک تهیه این مقاله بدون همکاری‌های گسترده و بی‌دریغ مسئولان شرکت اعوان به ویژه جناب آقای مهندس پیشوایی مقدور و میسر نبود که بدین وسیله کمال تقدیر و تشکر خود را از ایشان اعلام می‌داریم.

## چیدمان فصل‌های مقاله

این مقاله از ۸ فصل تشکیل شده است که در ادامه هر کدام از این فصول معرفی می‌شود.

در فصل اول، به تعریف اجمالی اصطلاحات مطرح در حوزه خط تولید پرداخته می‌شود که هدف آن آشنایی خواننده با مفاهیم این دامنه می‌باشد. منبع اصلی تهیه این فصل، کتاب رسمی درس [4] می‌باشد.

فصل دوم به معرفی مشروح شرکت اعوان، تاریخچه، گستردگی، محصولات و پروژه‌های انجام گرفته در این شرکت می‌پردازد که عمده مطالب این فصل برگرفته از سایت رسمی این شرکت [5] می‌باشد.

در فصل سوم به طور خاص بر روی معماری خط تولید شرکت اعوان که در قالب محصول wise عرضه شده است پرداخته می‌شود که بدین منظور از نمودارهای مختلف UML استفاده می‌شود. منبع این فصل، مستندات است که شرکت اعوان در اختیار نگارندگان قرار داده است و آشنایی مناسبی در مورد جنبه‌های فنی پیاده‌سازی خط تولید در عمل فراهم می‌کند.

در فصل‌های چهارم، پنجم و ششم ارزیابی میزان بلوغ خط تولید این شرکت مورد توجه قرار گرفته است. بدین منظور ابتدا در فصل چهارم مدل FEF [3] به طور کامل معرفی می‌شود. سپس در فصل پنجم پرسش‌نامه‌ای که برای اجرای این مدل توسط نگارندگان مقاله طراحی شده است، ذکر می‌شود. در فصل ششم نیز نتایج جلسات برگزار شده با مسئولان شرکت اعوان مطرح می‌شود. این جلسات از دید نگارندگان مقاله بسیار جلسات پربار و آموزنده‌ای محسوب می‌شود؛ بدین ترتیب مطالعه فصل ششم اکیدا توصیه می‌شود.

فصل هفتم مربوط به مرور مقالاتی [1,2] است که با رویکرد شرکت اعوان در زمینه خط تولید مطابقت زیادی دارد.

فصل هشتم نیز بر اساس جمع‌بندی‌های انجام شده به ارائه پیشنهاداتی برای بهبود خط تولید در شرکت اعوان می‌پردازد.

با تشکر مجدد از همکاری‌های شایسته و قابل تقدیر مسئولان شرکت اعوان، سند مطالعه موردی و ارزیابی خط تولید نرم‌افزاری شرکت اعوان در بیش از صد صفحه، تقدیم می‌گردد.



## ۱- تعریف اصطلاحات مرتبط

### ۱-۱- نیاز به خط تولید

بحث استفاده مجدد از فرآورده‌های مشابه پیشین، یکی از مباحث مهم در حوزه مهندسی نرم‌افزار است. با گسترده شدن تولیدات هر شرکت نرم‌افزاری، یکی از نکاتی که برای مدیران تیم‌های نرم‌افزاری و یا حتی توسعه‌دهندگان عادی نرم‌افزار قابل مشاهده است، وجود بخش‌های تکرارپذیر و مشابه در پروژه‌های مختلف است. بدین ترتیب در ساده‌ترین حالت، وجود یک انباره مشترک از مصنوعات تولید شده امری ضروری به نظر می‌رسد. اما این نکته تنها دلیل برای استفاده از خط تولید نیست؛ بلکه خط تولید مفهومی عام‌تر از این نکته در خود دارد. در خط تولید محصول تلاش می‌شود که بخش‌های مشترک از فرآیند توسعه محصولات مختلف نیز تا حد امکان یکپارچه شود.

در واقع خط تولید بر اساس دو مفهوم کلیدی بنا نهاده می‌شود: یکپارچه کردن داده‌ها، فرآیندها و تمامی مصنوعات مشترک و مشابه در یک بخش مرکزی و نیز تعریف نقاط تغییر مشخصی که هر کدام از محصولات مختلف در این نقاط می‌توانند مشخصه‌ای مختص به خود داشته باشند. یعنی توجه خاص به دو مفهوم شباهت (Commonality) و تفاوت (Variability).

### ۱-۲- مهندسی خط تولید نرم‌افزاری<sup>۲</sup>

«مهندسی خط تولید نرم‌افزاری پارادایمی است که برای توسعه برنامه‌های نرم‌افزاری (سامانه‌های متمرکز بر نرم‌افزار و محصولات نرم‌افزاری) از چارچوب‌ها و شخصی‌سازی گسترده استفاده می‌کند» [۴]. این تعریف مبتنی بر اصطلاحات «چارچوب» و «شخصی‌سازی گسترده» است که در ادامه معرفی می‌شوند.

### ۱-۳- چارچوب نرم‌افزاری<sup>۳</sup>

«یک چارچوب نرم‌افزاری مجموعه‌ای از زیرسیستم‌ها و واسط‌هاست که یک ساختار مشترک را ایجاد می‌کند تا مجموعه‌ای از محصولات مشتق‌گرفته شده بتوانند به صورت موثر توسعه داده شده و تولید شوند» [۴].

### ۱-۴- شخصی‌سازی گسترده<sup>۴</sup>

«شخصی‌سازی گسترده، تولید محصولات در حجم انبوه است که هر کدام برای نیازهای خاص مشتریان مناسب شده است» [۴].

<sup>۲</sup> Software Product Line Engineering

<sup>۳</sup> Software Platform

<sup>۴</sup> Mass Customization

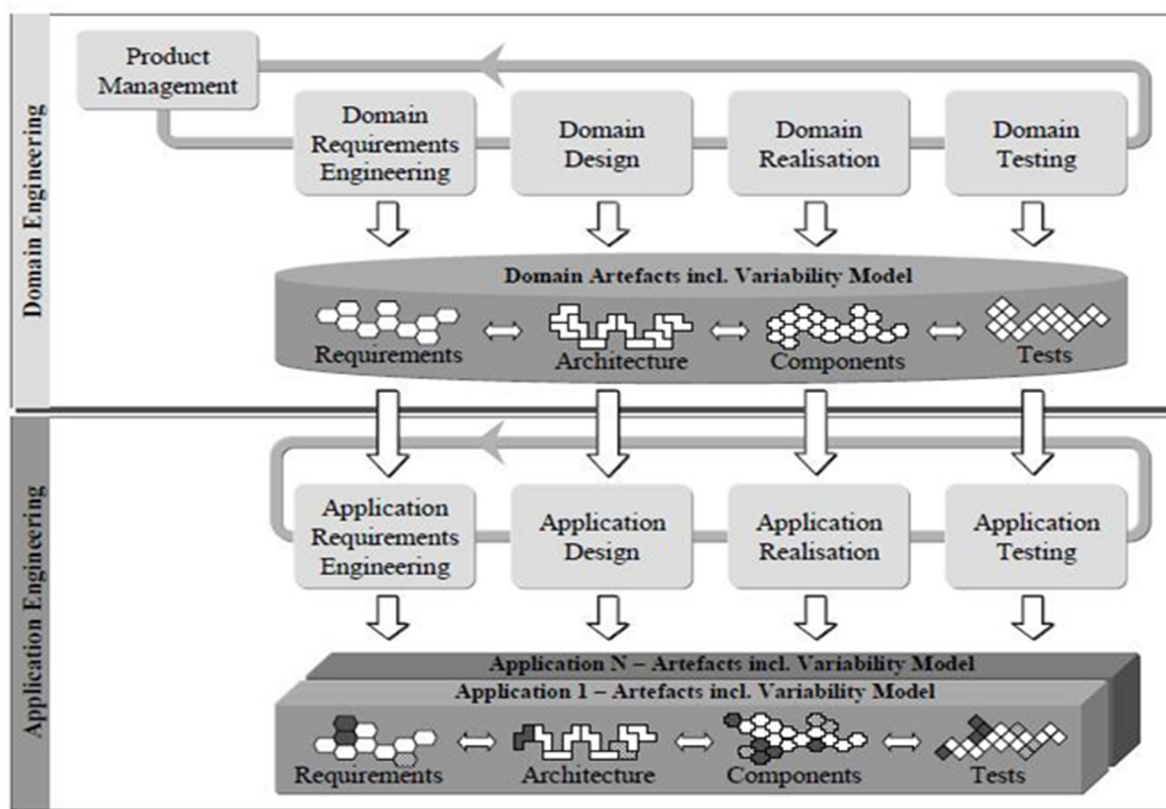
## ۱-۵- نقاط تغییر<sup>۵</sup> و متغیر<sup>۶</sup>ها

«هر نقطه تغییر، نمایشی از یک سوژه تغییرپذیری است که درون مصنوعات دامنه تعریف شده و توسط اطلاعات پیرامونی غنی شده است» [۴]. به بیان ساده‌تر، نقطه تغییر نقطه‌ای است که در حوزه دامنه تعریف می‌شود و امکان تعریف مقادیر متفاوتی را برای خود فراهم می‌کند.

«هر متغیر نمایشی از یک شی تغییرپذیری است که درون مصنوعات دامنه تعریف می‌شود» [۴]. مشابه تعریف قبل، متغیر در واقع مقداری است که به یک نقطه تغییر می‌توان نسبت داد.

## ۱-۶- چارچوب ارائه شده برای مهندسی خط تولید

یکی از چارچوب‌های مطرحی که برای مهندسی خط تولید نرم‌افزاری استفاده می‌شود در شکل ۱ نمایش داده شده است. این چارچوب بر اساس تفکیک مولفه‌های نرم‌افزاری به دو بخش مولفه‌های مطرح در دامنه و مولفه‌های مطرح در برنامه بنا نهاده شده است. در این چارچوب هر کدام از گام‌های چرخه نرم‌افزار نیز در دو بخش دامنه و برنامه صورت می‌گیرد.



شکل ۱ - چارچوب خط تولید نرم‌افزاری

<sup>۵</sup> Variation Point

<sup>۶</sup> Variant

به طور کلی این مدل بر این نکته کلیدی از مفاهیم خط تولید متمرکز دارد که به منظور بیشینه کردن امکان استفاده مجدد از مولفه‌ها، باید بخش‌های مشترک میان چند برنامه را که در یک دامنه خاص از کسب و کار مطرح می‌شوند، در قالب یک دسته از مصنوعات متمرکز کرد که به این دسته از محصولات، نام مصنوعات دامنه داده شده است. برای توسعه این دسته از مصنوعات، یک چرخه تولید مجزا در نظر گرفته می‌شود.

مصنوعاتی که خاص هر برنامه مستقل هستند، مصنوعات برنامه نامیده می‌شوند و برای تولید هر کدام از این برنامه‌ها، یک چرخه تولید دیگر در نظر گرفته خواهد شد.

### **۱-۷- جمع‌بندی**

در این فصل مفاهیم و اصطلاحات کلیدی مطرح در حوزه خط تولید نرم‌افزار بررسی شد. هدف از این فصل، آشنا ساختن خوانندگان با این مفاهیم و ارائه تعریفی یکسان از اصطلاحاتی است که در فصول آتی در مورد خط تولید به کار می‌رود و گاهی با عبارات مشابهی مانند core asset بیان می‌شود.

## ۲- معرفی شرکت اعوان

### ۲-۱- مقدمه

شرکت «مشاوران نرم‌افزاری اعوان» در سال ۱۳۸۳ تاسیس شده است. این شرکت عضو شورای عالی انفورماتیک و سازمان نظام صنفی رایانه‌ای می‌باشد. راهبرد اعوان ارائه خدمات فوق تخصصی با تکیه بر گردآوری نخبگان علمی و فنی کشور و افراد متعهد است.

در حال حاضر این شرکت صرفاً دارای یک شعبه در تهران می‌باشد ولی نیروهای آن در سازمان‌های مختلف به فراخور پروژه‌های مختلف، حضور پیدا می‌کنند.

### ۲-۲- محورهای اصلی فعالیت

#### ۲-۲-۱- تولید نرم افزارهای سفارشی

رویکرد اعوان در این حوزه کاهش هزینه و زمان تولید با به کارگیری خط تولید نرم‌افزار اعوان ( WISE Solution) و استفاده از تخصصی‌ترین فناوری‌ها همچون Java EE و Oracle می‌باشد.

#### ۲-۲-۲- خدمات آموزشی و مشاوره‌ای

رویکرد اعوان در این حوزه استفاده از اساتید کارکاشته (Practitioner) و حمایت علمی از دانش‌آموختگان طی یک فرایند مدت‌دار آموزش و به کارگیری مهارت‌ها است.

#### ۲-۲-۳- تولید بسته‌های نرم افزاری

رویکرد اعوان در این حوزه تولید محصولات تخصصی و راهگشا برای حل نیازمندی‌های واقعی سازمان‌ها و شرکت‌هاست.

### ۲-۳- مشتریان عمده

این شرکت توانسته است در مهمترین پروژه‌های بزرگترین سازمان‌های صنعتی یا اداری ایران همچون نهاد ریاست جمهوری اسلامی ایران (پروژه ملی سامد) و گروه صنعتی ایران خودرو نقش ایفا نماید. تعدادی از سازمان‌ها و ادارات مطرحی که این شرکت برای آن‌ها خدمات و محصولات ارائه داده است، به این شرح هستند:

نهاد ریاست جمهوری اسلامی ایران، مجلس شورای اسلامی، سازمان تامین اجتماعی، بنیاد شهید و امور ایثارگران، گروه صنعتی ایران خودرو، وزارت بهداشت درمان و آموزش پزشکی، مخابرات استان تهران، بانک

مسکن، شرکت انبارهای عمومی و خدمات گمرکی ایران، شرکت امداد خودرو ایران، سازمان ترافیک شهرداری تهران، شرکت پیک آسا، شرکت میکرو موج سیستم، شرکت فناوری اطلاعات پاسارگاد آرین (فناپ)، شرکت پیام همشهری، شرکت صنایع دارویی زردبند، وزارت مسکن و شهرسازی، وزارت امور خارجه.

## ۲-۴- محصولات نرم‌افزاری

### ۲-۴-۱- مدیریت ارتباط با شهروند

«مدیریت ارتباط با شهروند» یا citiZen Relationship Management - که به اختصار ZRM نامیده می‌شود - یک سیستم نرم‌افزاری یک‌پارچه است که پل ارتباطی سازمان با مراجعین را به صورت بهینه برقرار کرده و امر رسیدگی را در داخل سازمان تسهیل می‌نماید. بسیاری از سازمان‌ها، نهادها، بانک‌ها و موسسات با شهروندان ارتباط دارند و هنگامی که این رابطه در قالب درخواست و یا شکایت قرار می‌گیرد رنگ و بوی ویژه‌ای به خود گرفته، و از سوی سازمان و شهروند اهمیت خاصی می‌یابد.

از طرفی شهروند توقع دارد تا بتواند به راحتی با سازمان مربوطه ارتباط برقرار کرده و کار خود را به آن ارجاع دهد و در زمان مقتضی، پاسخ مناسبی دریافت نماید. از سوی دیگر سازمان برای تسریع و تصدیق امر رسیدگی نیازمند مکانیزه شدن فرآیندها و پرونده‌هاست تا به کاهش هزینه‌های بروکراسی اداری بتواند پاسخگوی نیاز شهروندان باشند. همچنین مدیران سطوح مختلف نیازمند سیستمی کنترلی برای رفع مشکلات و حتی پیش‌بینی و جلوگیری از وقوع حوادث آینده هستند تا با اقدامات به موقع و شناسایی گلوگاه‌های اصلی، اثر بخشی و کارایی را تا حد خوبی بهبود بخشند.

تسهیل ارتباط طرفین، رسیدگی سریع و دقیق به همراه پاسخ‌گویی درخور و شایسته، به همراه هاله‌ای از کنترل کل سیستم، نکات کلیدی این فرآیند پرمخاطره را تشکیل می‌دهند که با استفاده از ZRM، در دست مدیران توانمند این مهم چهره واقعی به خود می‌گیرد.

### ZRM برای شهروندان

- سهولت دسترسی به خدمات و حذف جغرافیا در سرویس‌دهی
- افزایش امکان خودخدمتی با استفاده از کانال‌های اینترنت و پیامک
- افزایش رضایت‌مندی در بهره‌گیری از خدمات استاندارد شده
- سهولت پیگیری خدمات مورد درخواست بدون مراجعه حضوری
- اطمینان از کیفیت خدمات در تعامل با سیستم ZRM
- بهره‌مندی از خدمت‌رسانی متوازن و عادلانه

## ZRM برای سازمان

- پاسخگویی دقیق، سازگار و کیفی با صرف هزینه و زمان کمتر
- ارزیابی عملکرد واحدهای سازمانی و مسئولین ذی ربط
- هدایت هوشمند فرایند رسیدگی به موارد
- توسعه دانش سازمانی برای بهبود خدمت‌رسانی در طول عمر سازمان
- دستیابی به سوابق مراجعات و پرونده خدمات شهروند از یک پنجره واحد
- بسترسازی برای ریشه‌یابی مشکلات موجود و پیش‌بینی نیازهای آتی
- افزایش رضایت شغلی کارمندان به دلیل تعامل سازمان یافته با ارباب رجوع

### ۲-۴-۲- پورتال سازمانی

سازمان‌ها و شرکت‌ها عموماً نیازمند مجموعه‌ای از سیستم‌ها برای پاسخگویی به نیازمندی‌های الکترونیکی خود هستند. پورتال درون‌سازمانی اعوان موسوم به WiSE Portal بستری مناسبی برای توسعه و تجمیع سیستم‌های مبتنی بر وب درون‌سازمانی فراهم می‌نماید به ترتیبی که کاربران با ورود به پورتال، از طریق یک پنجره از کلیه سیستم‌هایی که مجاز هستند استفاده نمایند. این محصول در بستر Java EE تولید شده است ولی امکان تجمیع با سیستم‌های وبی مبتنی بر PHP و Net. را نیز فراهم می‌نماید. انتقال سیستم‌های وبی موجود به این پورتال با تغییرات اندک امکان پذیر است.



شکل ۲ - پورتال سازمانی

## امکانات امنیتی

- مکانیزم هویت‌شناسی مجهز به Single Sign-on
- مدیریت کاربران و نقش‌ها
- تعریف کدهای دسترسی به ازای عملیات مختلف برای کلید سیستم‌ها در ساختاری درختی
- تعیین سطوح دسترسی بر اساس نقش‌ها و کدهای دسترسی
- گزارش‌دهی از فعالیتهای کاربران با ذکر زمان، IP، کد دسترسی و توضیح
- فعال‌سازی HTTPS
- امکان استفاده از Smart Card یا Security Token برای هویت‌شناسی
- تنظیم میزان پیچیدگی و طول عمر گذرواژه‌ها

## درگاه واحد کلید سیستم‌ها

- تعریف منو به تفکیک کلید سیستم‌ها با تعیین کد دسترسی هر منو
- نمایش درگاه ورودی کاربر با تجمیع کلید منوهای قابل دسترسی
- نمایش اطلاعیه‌ها و آلازم‌ها

## مدیریت اطلاعات ساختار سازمانی

- نگهداری اطلاعات ساختار سازمانی با ارتباطات درختی
- پشتیبانی از روابط ماتریسی بین سازمان‌ها با پیاده‌سازی نسخه بهینه‌سازی‌شده الگوی Accountability
- مدیریت اطلاعات سمت‌های سازمانی، انواع و سطح‌بندی سمت‌ها
- مدیریت سطوح محرمانگی
- تنظیم آیین‌نامه مکاتبات مابین انواع سمت‌ها

## سایر ویژگی‌ها

- تست شده بر سرورهای Tomcat، JBoss، WebLogic (قابل تطبیق برای سایر Application Server های استاندارد)
- تست شده بر روی سیستم‌عامل‌های Linux (Suse, Debian, Ubuntu, Fedora)، FreeBSD و Windows (XP, 2003) و پایگاه‌داده‌های Oracle، MySQL و SQL Server
- امکان برقراری Replication برای پایگاه داده‌های توزیع شده
- امکان برقراری Clustering در لایه پایگاه داده و Application Server
- امکان نصب و تنظیم از طریق ابزار نصب خودکار

## ۲-۴-۳- خط تولید نرم افزار (WiSE Solutions)

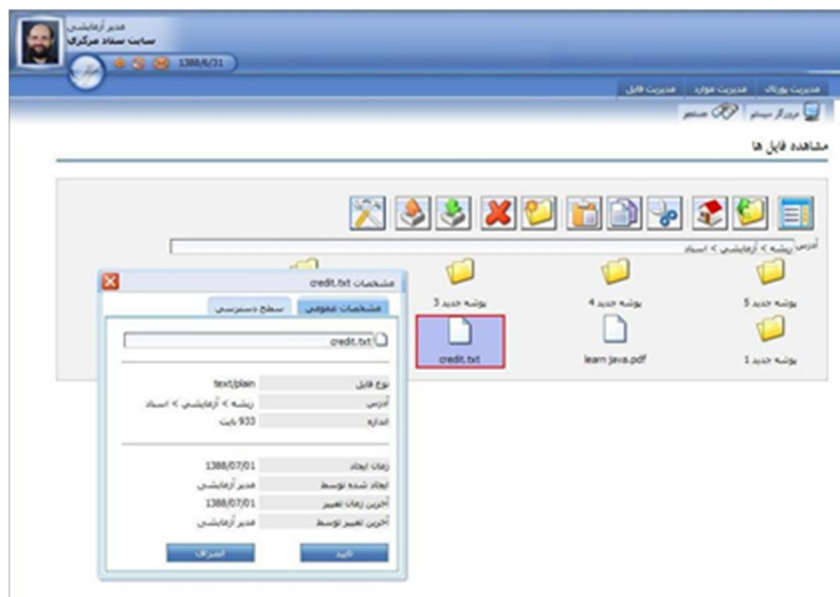
شرکت اعوان با کنار هم قرار دادن مجموعه‌ای از محصولات زیرساختی خود به یک خط تولید نرم افزار دستیافته است. این محصول به وسیله کارشناسان اعوان و شرکتهای همکار در پروژه‌های بسیار زیادی مورد استفاده قرار می‌گیرد. این راه حل شامل ماژول‌های زیر می‌باشد:

- WiSE Core : شامل کلاس‌ها و سایر منابع زیرساختی برای تسهیل پیاده‌سازی موارد کاربرد
- WiSE Code Generator : ابزار تولید کد برای سیستم‌های داده محور بر اساس اطلاعات مدل داده
- WiSE Workflow : اضافه کردن گردش کار به سیستم‌ها با استفاده از استاندارد XPD

محصولات مبتنی بر WiSE Core با قرار گرفتن کنار محصولاتی شبیه به WiSE Portal، WiSE RMS و WiSE Installer می‌توانند زمان تولید و راه‌اندازی سیستم‌های داده محور را تا بیش از ۵۰ درصد کاهش دهند.

## ۲-۴-۴- سیستم مدیریت اسناد (DBFS)

سیستم مدیریت اسناد اعوان موسوم به DBFS امکاناتی شبیه به Windows File Explorer را در محیط وب فراهم می‌کند. این سیستم که در بستر Java EE و مبتنی بر WiSE تولید شده است با ذخیره اطلاعات فایل‌ها در پایگاه داده جستجو، پشتیبان گیری، Replication و سایر امور مربوط به ذخیره و بازیابی فایل‌ها را تسهیل می‌نماید.



شکل ۳ - سیستم مدیریت اسناد



## امکانات

- ذخیره و بازیابی و تغییر انواع فایل
- فولدربندی فایل‌ها
- مشاهده فایل‌ها در دو حالت‌های Icon View و Detail View
- امکان جستجو بر اساس نام، مشخصات، کلیدواژگان
- جستجو در محتویات فایل‌های Text، PDF، Word، HTML و Excel
- نگهداری سوابق تغییرات و نسخ فایل‌ها
- مدیریت دسترسی روی فایل‌ها و فولدرها
- دریافت مستقیم فایل از اسکنر (با استفاده از Applet)
- دریافت دسته‌ای فایل از خروجی اسکنرهای صنعتی (تحت شبکه)
- انتقال فایل‌ها از طریق عملیات Copy، Cut و Paste
- تعریف کلیدهای میانبر مشابه Windows Explorer برای عملیاتی همچون Copy، Cut، Paste
- تعریف گردش کار برای تعیین روند بررسی و بازبینی فایلها
- نمایش کارتابل فایل‌های ارجاع شده به هر کاربر

### ۲-۴-۵ - سیستم مدیریت گزارش‌های پویا

انعطاف‌پذیری در طراحی و تولید گزارشات یکی از دغدغه‌های اصلی مدیران سازمانهاست. با سیستم مدیریت گزارشات پویای اعوان موسوم به WiSE RMS می‌توانید گزارشاتی را که بر روی پایگاه داده و جداول خود با استفاده از موتورهای مشهور و استاندارد گزارش‌سازی همچون Crystal Reports یا BIRT یا JasperReports تعریف می‌شوند در اختیار مدیران و کاربران قرار داده می‌شود.

ID	نام	تاریخ	وضعیت
1	محمد احمدی	1388/05/10	موجود
2	علی محمدی	1388/05/11	موجود
3	سید علی حسینی	1388/05/12	موجود
4	مهدی احمدی	1388/05/13	موجود
5	فرمان احمدی	1388/05/14	موجود
6	علیرضا احمدی	1388/05/15	موجود
7	مهدی احمدی	1388/05/16	موجود
8	علیرضا احمدی	1388/05/17	موجود
9	فرمان احمدی	1388/05/18	موجود
10	مهدی احمدی	1388/05/19	موجود
11	فرمان احمدی	1388/05/20	موجود
12	علیرضا احمدی	1388/05/21	موجود
13	فرمان احمدی	1388/05/22	موجود
14	علیرضا احمدی	1388/05/23	موجود
15	فرمان احمدی	1388/05/24	موجود
16	علیرضا احمدی	1388/05/25	موجود
17	فرمان احمدی	1388/05/26	موجود
18	علیرضا احمدی	1388/05/27	موجود
19	فرمان احمدی	1388/05/28	موجود
20	علیرضا احمدی	1388/05/29	موجود
21	فرمان احمدی	1388/05/30	موجود
22	علیرضا احمدی	1388/05/31	موجود
23	فرمان احمدی	1388/06/01	موجود
24	علیرضا احمدی	1388/06/02	موجود
25	فرمان احمدی	1388/06/03	موجود

شکل ۴ - سیستم مدیریت گزارش

## امکانات

- طراحی گزارش از روی پایگاه داده با استفاده از ابزارهای استاندارد طراحی گزارش متنوع همچون iReport و Crystal Report Designer، BIRT Designer
- اجرای گزارشات با موتورهای گزارش سازی متنوع
- بارگذاری گزارشات و تعیین سطوح دسترسی به هر گزارش
- تعریف پارامترهای ورودی برای اجرای هر گزارش از انواع مختلف (رشته‌ای، عددی، تاریخ، انتخاب و ...)
- مشاهده لیست گزارشات توسط کاربر و انتخاب و اجرای گزارشات با کنترل دسترسی
- زمان بندی اجرای گزارشات
- ارسال خروجی گزارش از طریق ایمیل

### ۲-۴-۶- PetroViz

فعالان صنعت نفت به خوبی به ارزش میلیارد دلاری داده‌ها نفتی واقفند. این داده‌ها می‌توانند مبنای تصمیم‌گیریهای درست برای افزایش میزان استحصال نفت و در نتیجه درآمدهای نفتی باشند. نظر به امکان پذیر نبودن بهره‌برداری از نرم‌افزارهای گران و سنگین نفتی همچون Roxar RMS یا Petrel روی دستگاه‌های کلیه متخصصین شرکت اعوان با تولید PetroViz امکان مطالعه و پژوهش روی این داده‌ها بدون نیاز به نصب نرم‌افزارهای جانبی را فراهم آورده است. این محصول امکان نمایش احجام، نمودارها و رویه‌ها، گریدها و سایر داده‌های نفتی را فراهم می‌کند.

### ۲-۴-۷- ابزار نصب و به‌روزرسانی (WiSE Installer)

نصب و به‌روزرسانی سیستم‌های مبتنی بر Java EE از دغدغه‌ها و مشکلات راهبران این سیستم‌ها می‌باشد. ابزار نصب و به‌روزرسانی اعوان موسوم به WiSE Installer شما را یاری می‌کند تا برنامه‌های Java EE را به سادگی طی چند دقیقه نصب و تنظیم کنید و نگران فراموش شدن بخش‌هایی از تنظیمات و عدم اجرای صحیح قسمت‌هایی از عملیات نباشید. همچنین به‌روزرسانی نسخ جدید روی سرورها به ویژه زمانی که بیش از یک سرور در حال سرویس‌دهی باشند و کنترل هماهنگی نسخ یک دیگر از مشکلاتی است که این ابزار به‌روزرسانی آن را به صورت خودکار حل می‌نماید.

## امکانات

- نصب خودکار برنامه‌ها روی سرورهای JBoss
- خودکارسازی نصب و تنظیم برنامه‌ها روی Web Logic با استفاده از Jython و WLST

- توسعه‌پذیری ابزار نصب با اضافه کردن plug-in های جدید
- ارسال و دریافت خودکار نسخ جدید نرم‌افزار از طریق FTPS و نصب اتوماتیک نسخ جدید

#### ۲-۴-۸- ابزار پایگاه داده (WiSE DB Tools)

ابزار پایگاه داده شرکت اعوان، امکاناتی را برای کار با پایگاه داده‌های اوراکل در اختیار می‌گذارد.

#### امکانات

- ایجاد اسکریپت‌های جریان اوراکل که برای برقراری مکانیزم‌های تکرار شرطی برپایه XML استفاده می‌شود.
- مدیریت پیشرفته تداخل (conflict) برای رفع تناقضات حذف و به‌روزرسانی نوع‌های مختلف
- چارچوب تست برای موارد تست تکرار

#### ۲-۴-۹- جدول همراه

جدول همراه محصول شرکت اعوان، امکان حل جداول کلمات متقاطع و جداول اعداد را بر روی گوشی‌های مجهز به Java فراهم می‌کند.



#### ۲-۴-۱۰- پرتال اینترنتی

پرتال شرکت اعوان، نرم‌افزار مدیریت محتوای وبسایت، با امکانات معمول نرم‌افزارهای تحت است که می‌توان با آن پرتال اینترنتی مورد نیاز شرکت‌ها و سازمان‌ها را راه‌اندازی کرد. این پرتال در کنار پرتال درون‌سازمانی به سرویس‌دهی بهتر و روانتر به مشتریان و مراجعین کمک می‌نماید. شرکت اعوان علاوه بر راه‌اندازی پرتال مورد نیاز، آماده است برای تهیه محتوا و مطالب سایت نیز شرکت یا سازمان را یاری می‌کند. این پرتال، مکانیزم‌های امنیتی WiSE را پیاده می‌کند و همچنین یک راه ورود و نمایش اطلاعات یکپارچه را در اختیار می‌گذارد.

## امکانات

- امکانات و ویژگی‌های معمول و عمومی نرم‌افزارهای تحت وب
- پیاده‌سازی مکانیزم‌های امنیتی نرم‌افزارهای تحت وب، شامل صفحه ورود، مدیریت کاربری، کنترل دسترسی و گزارش وقایع (log) فعالیت‌های کاربران
- دروازه یکپارچه برنامه‌ها که امکان ورود یک‌باره و دسترسی به تمام برنامه‌ها (به شکل منو) را در اختیار گذاشته و نمای یکپارچه از تمام برنامه‌ها را نمایش می‌دهد.
- مدیریت اطلاعات ساختار سازمانی
- API برای پرسش اطلاعات کاربران
- مدیریت اطلاعات برنامه‌ها، سرورها و مراکز داده نصب شده در شبکه که در حالت استفاده از راه دور، به‌خصوص در معماری‌های توزیع شده، کاربرد دارد.

## ۲-۵- سایر خدمات شرکت اعوان

### ۲-۵-۱- تولید نرم‌افزار سفارشی

شرکت اعوان با اتکا بر خط تولید نرم‌افزار خود موسوم به WISE Solutions سعی در تولید سریع، با کیفیت و کم‌هزینه نرم‌افزارهای سفارشی دارد. عمده تولیدات این شرکت روی بستر Java EE و مبتنی بر وب می‌باشند. طیف وسیع و متنوع این تولیدات شامل سیستم‌های مدیریت خدمات، سیستم‌های CRM، مدیریت گردش کار، پرتال‌های و سایت‌های اینترنتی پر بیننده، آزمون‌های الکترونیکی، کنسول‌های مدیریت و مونی‌تورینگ سرویس‌های مخابرات، سیستم Billing برای مراکز تلفن و ... را شامل می‌شود.

### ۲-۵-۲- پشتیبانی

شرکت اعوان جهت افزایش دسترس‌پذیری، کارایی، امنیت و افزایش کیفیت بهره‌برداری پشتیبانی سرورهای Oracle و Web Logic و JBoss را به عهده می‌گیرد و اقدام زیر را در این راستا مدیریت می‌کند:

- راه‌اندازی Oracle Replication
- راه‌اندازی Oracle Data Guard
- راه‌اندازی Oracle Clustering
- تنظیم ابزارهای Backup و Recovery
- راه‌اندازی Application Server Load Balancer
- راه‌اندازی Application Server Cluster
- راه‌اندازی ابزارهای مونی‌تورینگ سرورها و وضعیت سرویس‌ها

## ۲-۵-۳- مشاوره

تولیدکنندگان نرم‌افزار به ویژه شرکت‌هایی استفاده کننده از خط تولید نرم‌افزار اعوان می‌توانند با بهره‌گیری از خدمات مشاوره‌ای کارشناسان اعوان مسیر تولید خود را هموارتر نمایند. این مشاوره با اهداف زیر صورت می‌پذیرد:

- تدوین معماری نرم‌افزار
- راه‌اندازی بستر تولید نرم‌افزار در آغاز پروژه
- استفاده از فناوری‌های Java EE و Java

## ۲-۵-۴- نظارت

شرکت‌ها و سازمان‌هایی که تمایل به سفارش بسته‌ها یا تولیدات نرم‌افزاری دارند می‌توانند از خدمات نظارتی و مشاوره‌ای اعوان بهره بگیرند. این خدمات شامل موارد ذیل خواهد بود:

- تهیه یا بازبینی اسناد مناقصه
- ارزیابی محصولات یا مناقصه‌گران
- ارزیابی ساختاری و محتوایی مستندات تحویل شده مجری
- ارزیابی پارامترهای کیفی نرم‌افزار همچون امنیت و کارایی

## ۲-۵-۵- برون‌سپاری

شرکت اعوان آمادگی دارد مسئولیت تحلیل، طراحی، تولید و پشتیبانی یک بخش یا تمام بخش‌های قراردادهای نرم‌افزاری شرکت‌های بزرگ را به عهده بگیرد و با همکاری و تعامل اقدام به تحویل نرم‌افزار به Prime Contractor یا کارفرمای اصلی نماید.

## ۲-۵-۶- آموزش

شرکت اعوان دوره‌های آموزشی در محل شرکت برای افراد علاقمند و یا به سفارش سازمانها و شرکتهای در محل ایشان و به صورت کارگاهی برگزار می‌کند. همچنین نفرات برگزیده در هر دوره را در صورت علاقمندی فرد، جهت همکاری به چند شرکت معتبر طرف قرار داد، معرفی و پیشنهاد می‌دهد. دوره‌های آموزشی ارائه شده توسط اعوان به صورت زیر است:

### آموزش زبان برنامه‌نویسی جاوا

- فراگیری مهارت‌های لازم برای برنامه‌نویسی جاوا و مقدمات لازم برای مشغول به کار شدن در این بستر

- آموزش Java EE در Web Development
- آشنایی با Java EE، آموزش JSP، Servlet، JSTL، Spring
- آموزش برنامه‌نویسی Java روی موبایل
- آشنایی با Java ME و MIDP و CLDC

### آموزش اوراگل

- آموزش Oracle Administration (دوره‌های اوراگل فقط برای شرکت‌ها و سازمان‌ها و یا خصوصی برگزار می‌شود و ثبت نام عادی ندارد)
- آشنایی با معماری و مفاهیم، راه‌اندازی و تنظیمات، مدیریت کاربران، جداول و داده‌ها، امنیت، Performance، Backup و Recovery، PL/SQL، SQL Developer

### مجموعه آموزشی

اوراگل	مهندسی نرم افزار	جاوا
Oracle Streams Replication	Patterns	Hibernate
Oracle Clustering	Refactoring	JSF
	Load Test	GWT
	Unit Testing	Spring Framework
	Profiling Tools	Groovy
	UML	JBossRules
		Spring Security
		Web Services

## ۳- معرفی معماری خط تولید شرکت اعوان

### ۳-۱- مقدمه

خط تولید شرکت اعوان در قالب پروژه‌ای به نام WISE تجلی یافته است. هدف از این پروژه، متمرکز کردن مولفه‌های مختلفی است که در پروژه‌های گوناگون توسعه یافته‌اند. در این بخش، معماری این محصول به صورت مفصل مورد بررسی قرار می‌گیرد و از نمودارهای UML برای نمایش ساختار و ارتباطات میان اجزای پروژه core و سایر برنامه‌های نرم‌افزاری استفاده می‌شود.

### ۳-۲- معرفی wise

هر برنامه‌نویس با تجربه‌ای که اقدام به تولید سیستم‌های Enterprise مبتنی بر وب کرده باشد قطعاً با دغدغه‌هایی عمومی همچون امنیت، چندزبانگی، ایجاد و استفاده از سرویس‌های وب، تولید گزارشات، زمان‌بندی کارها و بسیاری از دغدغه‌های ریز و درشت دیگری که تقریباً در تمامی پروژه‌های این چنینی مشترک هستند آشنایی دارد.

هر تیم برنامه‌نویسی برای تسهیل کار خود به یک معماری منسجم، منعطف، پوشا، پویا و پایدار برای پاسخگویی به این نیازمندی‌های عمومی نیاز دارد و تنها در صورت وجود این سرمایه ارزشمند می‌تواند با سرعت و کیفیت قابل قبول اقدام به تولید نماید.

بستر تولید نرم‌افزار Java EE کوشیده است فضایی فراهم نماید که در آن عمده دغدغه‌های عمومی نرم‌افزارهای Enterprise به خوبی پاسخ داده شود. به دلیل تعدد راه‌حل‌ها، تعدد پیاده‌سازی استانداردها، تنوع امکانات و پراکندگی سرویس‌دهندگان، طراحی و تولید شالوده نرم‌افزاری پایدار این چنینی در بستر Java EE نیازمند تجربه و تسلط بسیار زیاد می‌باشد.

شرکت مشاوران صنعت نرم‌افزار اعوان کوشیده است شالوده نرم‌افزاری خود را که بر اساس بستر Java EE بنا شده است در قالب مجموعه‌ای از محصولات تحت عنوان WISE Solution ارائه دهد. کلمه WISE مخفف Web-based Information System Elite می‌باشد. همانگونه که از نام آن پیداست WISE صرفاً به حل دغدغه‌های سیستم‌های مبتنی بر وب می‌پردازد.

استفاده از WISE به برنامه‌نویس اجازه می‌دهد فارغ از دغدغه‌های عمومی با سرعت، موارد کاربرد مربوط به پروژه خود را به انجام برساند.

- از راه‌حل wise در موارد زیر استفاده می‌شود:

- تحلیل و طراحی
  - مدل سازی نیازمندی های استاندارد شده
  - معماری استاندارد شده
- توسعه
  - استفاده از فناوری های متن باز
  - تنظیمات و بهم دوختن فناوری ها
  - پیاده سازی الگوهای سازمانی
  - کاهش کارهای تکراری
  - ساز و کار ساخت پروژه
- تولید
  - مولفه های آماده برای تولید
  - ابزارهای نصب و به روز رسانی
  - ابزارها و الگوهای آزمون بار
- آموزش و مدیریت دانش
  - مستندات
  - تالار گفتگو

### ۳-۳- امکانات چارچوب wise

امکانات WISE در چهار محور قابل دسته بندی است:

- ۱- امکانات زیرساختی: شامل کلاسها، کتابخانه ها و سایر منابع همچون صفحات JSP، فایل های پیکربندی، دستورات ساخت، تصاویر و ... که در عمده پروژه ها برای پیاده سازی موارد کاربرد مفید و ضروری هستند ولی خود هیچ مورد کاربردی را پیاده نمی کنند. این منابع در زمان اجرای برنامه به عنوان بخشی از مولفه های نهایی ظاهر می شوند.
- ۲- برنامه های عمومی: مولفه های قابل اجرایی که موارد کاربرد عمومی مانند Login، منوها، مدیریت کاربران، ساختار سازمانی، کارتابل و سرویسهای عمومی مانند هویت شناسی و مجازشناسی را پیاده سازی می کنند. این برنامه های در زمان اجرا در قالب مولفه های مستقل در Application Server نصب می شوند.
- ۳- ابزارهای تولید: ابزارهای که در زمان توسعه سیستم به برنامه نویس کمک می کنند مانند ابزارهای تولید کد.



۴- ابزارهای پشتیبانی: ابزارهایی که در زمان بهره‌برداری به پشتیبان سیستم یاری می‌رسانند مانند ابزارهای نصب و به روزرسانی.

امکانات زیرساختی wise از موارد زیر تشکیل شده است:

- **GFaces: org.gfaces** که کلاس‌های پایه و ویجت‌های مشترک مربوط به فناوری gwt در آن ذخیره می‌شود
- **WiSE Core: ir.asta.wise.core** که کلاس‌های پایه و موارد مرکزی مانند امنیت، تست، اتصال از راه دور، گزارش‌گیری، نمایش، تراکنش‌ها و ثبت در پایگاه داده در این بخش قرار می‌گیرد
- **WiSE Lib: ir.asta.wise.lib** که توابع کتابخانه‌ای مورد نیاز بر اساس ساختار بسته‌بندی maven در قالب مخازن داده‌ای ذخیره می‌شوند.

در بخش برنامه‌های عمومی موارد زیر وجود دارند:

- **Portal: ir.asta.wise.portal**
- **Activiti: ir.asta.wise.activiti**
- **KM: ir.asta.wise.km**

ابزارهای تولید نیز شامل موارد زیر است:

- **Code Generator: ir.asta.wise.codegen**
- **Model Editor: ir.asta.wise.codegen.editor**

### ۳-۴ - ساختار برنامه‌های توسعه داده شده با wise

برنامه‌های توسعه داده شده با wise بر روی Application Server و در قالب تعدادی مولفه وب<sup>۷</sup> نصب می‌گردد. برخی از این مولفه‌ها از نوع برنامه‌های عمومی<sup>۸</sup> هستند و برخی از نوع سیستم‌های عملیاتی<sup>۹</sup>.

برنامه‌های عمومی سیستم‌هایی هستند که نیازمندیهای عمومی در سیستم‌های اطلاعاتی را پوشش می‌دهند همچون مدیریت کاربران یا مدیریت گردش کار.

---

<sup>۷</sup>Web Component

<sup>۸</sup>Fundamental

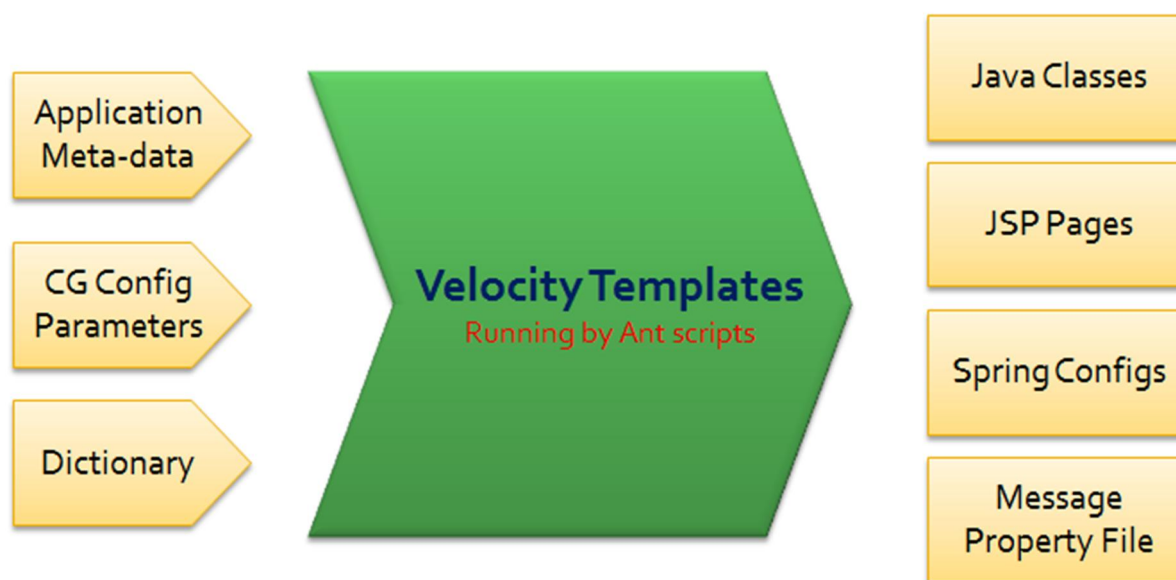
<sup>۹</sup>Operational Web Component

سیستمهای عملیاتی سیستمهایی هستند که به یک منظور خاص و برای پیادهسازی موارد کاربرد و قابلیتهای اصلی نرم افزارها تولید می شوند همچون سیستم مکاتبات، سیستم ارتباط با مشتری و آموزش الکترونیک.

سیستم پرتال (portal) از مولفه های کلیدی است که سیستمهای عملیاتی از آنها سرویس می گیرند.

دو پروژه پایه ای در WISE وجود دارند که برای تولید و اجرای هر یک از برنامه های مبتنی بر WISE به آنها نیاز است. پروژه های پایه ای یعنی wise.core و wise.lib به تنهایی هیچ مورد کاربردی را پیاده سازی نمی کنند و تنها پیاده سازی موارد کاربرد را در پروژه های دیگر تسهیل می کنند. زمانی که یک پروژه deploy می شود از منابع موجود در core و lib نیز بهره می گیرد و کلاس ها و سایر فایل های مورد نیاز از آنها را در application server کپی/نصب می کند.

تولید کدهای زمان اجرا در پروژه ها با استفاده از ابزار velocity صورت می گیرد که در شکل ۵ نمایش داده شده است:



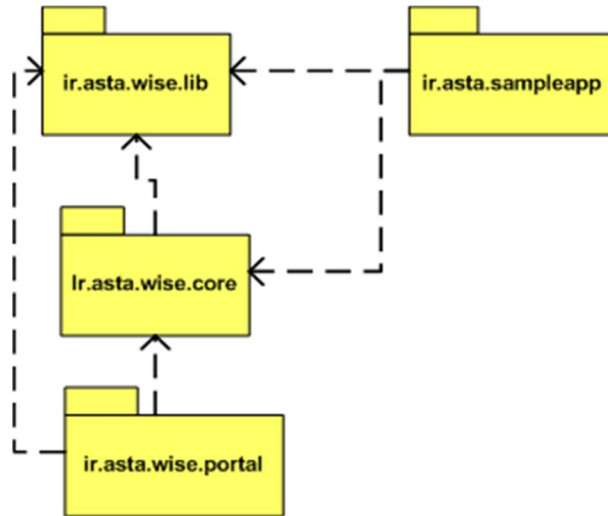
شکل ۵ - نقش Velocity در تولید کد

در توضیح این تصویر موارد زیر قابل ذکر است:

- ابزار velocity یک موتور پردازش قالب مبتنی بر جاوا مشهور و ساده است.
- ساختار Application Meta-data یک قالب مبتنی بر xml و خاص چارچوب wise است.
- بخش Model Editor یک افزونه eclipse مبتنی بر EMF برای ویرایش متاداده های برنامه هاست.
- چارچوب EMF یک چارچوب مدل سازی و تولید کد است

### ۳-۵- معرفی ارتباط مولفه‌های core و سایر برنامه‌ها

در Package Diagram شکل ۶، به صورت دقیق ارتباط میان مولفه‌های core که نقش دارایی‌های مشترک را دارند با مولفه‌های برنامه‌های عادی نمایش داده شده است:

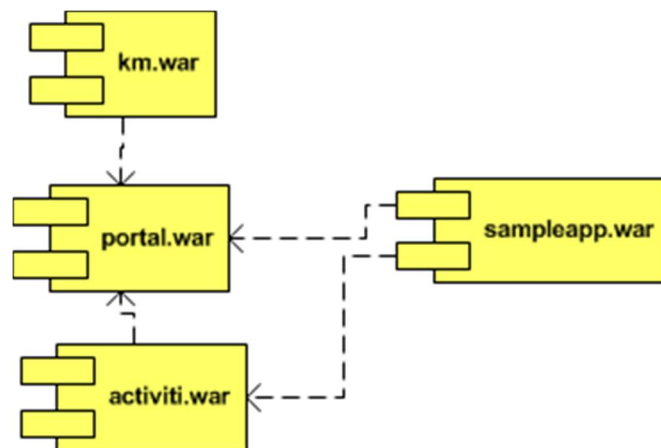


شکل ۶ - ارتباط بسته‌های core و برنامه

پروژه‌های منبع از بسته‌های زیر تشکیل شده‌اند:

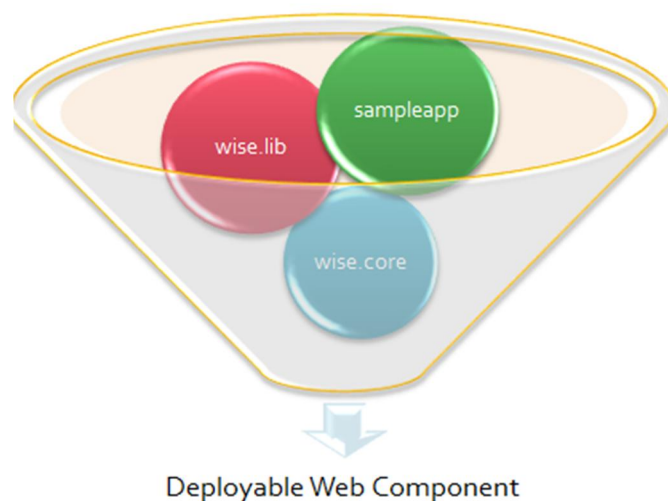
- ir.asta.wise.lib
- ir.asta.wise.core
- ir.asta.wise.portal
- ir.asta.wise.activiti
- ir.asta.wise.km
- ir.asta.sampleapp

در Component Diagram شکل ۷ ارتباط میان مولفه‌های core و برنامه‌ها نشان داده شده است:



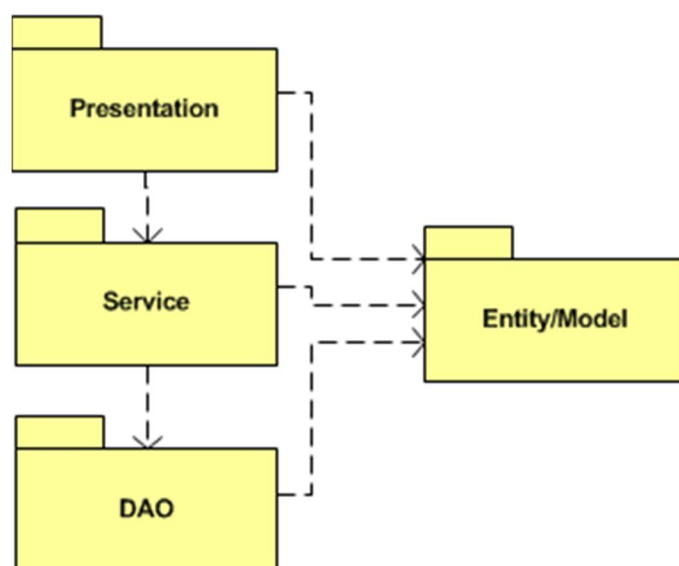
شکل ۷ - ارتباط مولفه‌های core و برنامه در زمان نصب

مکانیزم ساخت برنامه‌ها به کمک اسکریپت‌های ant صورت می‌گیرد. در این اسکریپت، مولفه‌های هر برنامه به صورت جداگانه کامپایل می‌شود و سپس در کنار مولفه‌های core قرار می‌گیرد. در نتیجه در برنامه وب نهایی، تمامی مولفه‌های مورد نیاز وجود خواهند داشت. در شکل ۸ این مفهوم نشان داده شده است:



شکل ۸ - قرار گرفتن مولفه‌ها در زمان نصب

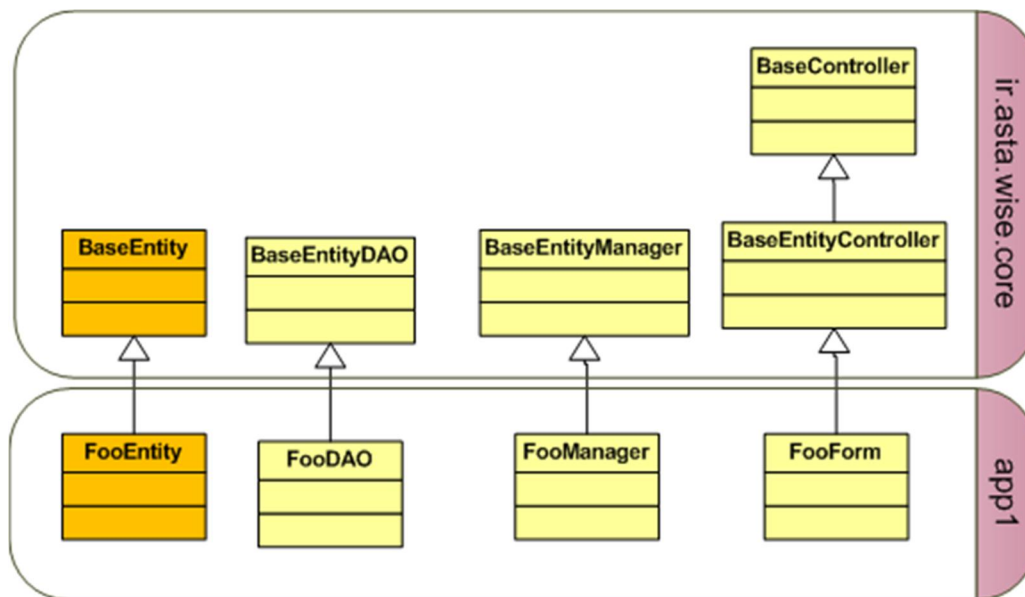
معماری برنامه‌های توسعه داده بر اساس چارچوب wise از سبک معماری لایه‌ای تبعیت می‌کند که در شکل ۹ این لایه‌ها نمایش داده شده‌اند:



شکل ۹ - نمودار لایه‌ای مولفه‌های معماری

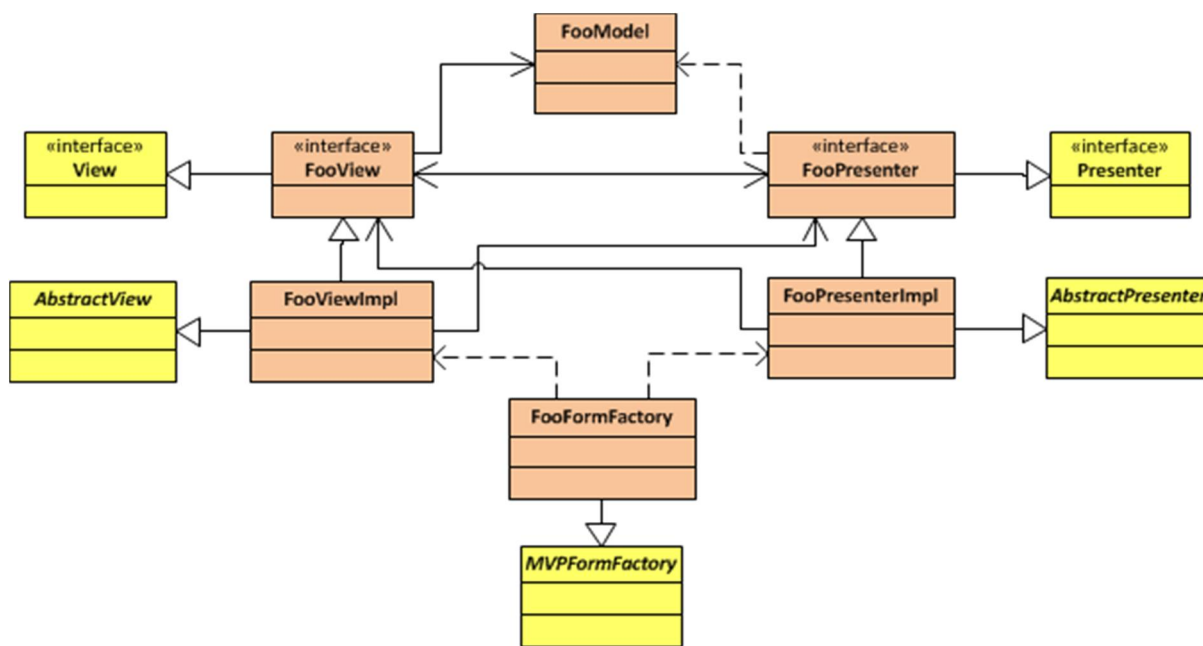
در Class Diagram شکل ۱۰ ارتباط میان بسته‌های هر لایه در یک برنامه نمونه و مولفه‌های متناظر در core در لایه دیتابیس نشان داده شده است. همان‌گونه که مشاهده می‌شود، قابلیت‌های مشترک در لایه

core تعریف می‌شود و سپس با استفاده از مفهوم وراثت، در کلاس‌های فرزند امکان استفاده از قابلیت‌های لایه core فراهم می‌شود. به علاوه هر کدام از موجودیت‌های مطرح در لایه app، می‌توانند خصوصیات مختص به خود را اضافه کنند.



شکل ۱۰ - ارتباط کلاس‌های core و برنامه

در Class Diagram شکل ۱۱ ارتباط میان کلاس‌های core و برنامه برای تولید ویجت‌های gwt نشان داده شده است:



شکل ۱۱ - ارتباط کلاس‌های core و برنامه در gwt

### ۴-۱- مقدمه

«مهندسی خانواده محصولات نرم‌افزاری»<sup>۱۰</sup> یک رویکرد راهبردی است که بر روی کسب و کار، سازمان و فناوری تاثیر می‌گذارد. این رویکرد به عنوان راهی برای توسعه متنوع محصولات نرم‌افزاری و سامانه‌های مبتنی بر نرم‌افزار با هزینه‌های پایین، در زمان کوتاه و با کیفیت بالا شناخته شده است. گزارش‌های تجربی متعددی، دستاوردهای پرمعنا به دست آمده را به وسیله معرفی خانواده محصولات در صنعت نرم‌افزار مستندسازی کرده‌اند.

### ۴-۱-۱- تاریخچه FEF

در بازه سال‌های ۱۹۹۵ تا ۲۰۰۵ میلادی، یک سری از پروژه‌های مشارکتی در زمینه مهندسی خانواده محصولات نرم‌افزاری انجام شد. در این سند یکی از نتایج مستحکم این سری از پروژه‌ها گزارش داده می‌شود. این سری با ESPRIT پروژه ARES شروع شد. مابین سال‌های ۱۹۹۵ تا ۱۹۹۸، سه شرکت اروپایی و سه دانشگاه با یکدیگر در زمینه معماری خانواده نرم‌افزارها همکاری کردند. مابین سال‌های ۱۹۹۸ تا ۲۰۰۰، ESPRIT پروژه PRAISE مشکلات فرآیندی مهندسی خانواده محصولات نرم‌افزاری را مورد بازرسی قرار داد. بر پایه این نتایج، یک گروه از صنایع اروپایی با یکدیگر و به همراه یک مجموعه از موسسات تحقیقاتی و سازمان‌های کوچک و متوسط در این زمینه همکاری کردند که از سال ۱۹۹۹ در یک سری از پروژه‌های درون چارچوب ITEA قرار گرفت. این پروژه‌ها شامل پروژه‌های ESAPS، CAFE و FAMILIES محسوب می‌شوند. شرکت‌هایی که درگیر این پروژه‌ها بودند بر روی یک گستره متنوع از سامانه‌های نهفته کار می‌کردند که شامل تصویربرداری پزشکی، گوشی‌های تلفن همراه، نرم‌افزار کنترل پرواز، کنترل ابزار، سرپرستی و مدیریت، خدمات مالی و تجهیزات الکترونیکی ماشین‌ها به شمار می‌رفت.

دو تمایز اصلی مابین مهندسی خانواده محصولات نرم‌افزاری و مهندسی یک سامانه نرم‌افزاری منفرد در این است که استفاده مجدد مدیریت شده از طریق جداسازی توسعه سیستم به دو بخش مهندسی دامنه و مهندسی برنامه صورت می‌گیرد و نیز تفاوت بعدی به مدیریت صریح تغییرپذیری<sup>۱۱</sup> برمی‌گردد. مهندسی دامنه، دارایی‌های دامنه را تولید می‌کند که در آن تغییرپذیری برای استفاده مجدد در نظر گرفته شده است. مهندسی برنامه، سامانه‌ها را با استفاده مجدد از متغیر<sup>۱۲</sup> های دارایی‌های دامنه ایجاد می‌کند.

Software Product Family Engineering<sup>۱۰</sup>  
variability<sup>۱۱</sup>  
variant<sup>۱۲</sup>

در پروژه ESAPS فرآیند توسعه و مدیریت تغییرپذیری از دیدگاه معماری و کیفیت مورد بازرسی قرار گرفت. در پروژه CAFE دغدغه‌های کسب و کار، نیازمندی‌ها، مدیریت دارایی‌ها و آزمون بیشتری معرفی شد. در پروژه FAMILIES نتایج و کارهای برجسته موجود در نتایج ESAPS یکپارچه شد و مورد ارتقا و خودکارسازی از طریق مهندسی خانواده مبتنی بر مدل و استفاده مجدد در مرزهای خانواده قرار گرفت.

#### ۴-۱-۲- ارتباط FEF با سایر رویکردها

چارچوب FEF تنها چارچوبی نیست که برای ارزیابی یا بنیان‌گذاری توسعه نرم‌افزار وجود دارد. به طور خاص، در حوزه فرآیندهای توسعه نرم‌افزار، مدل‌های ارزیابی قابلیت متعددی وجود دارند. این مدل یک دید انتزاعی از فرآیند توسعه نرم‌افزار ارائه می‌دهد. فعالیت‌های واقعی در یک سازمان به مقادیر انتزاعی نگاشت می‌شوند و از شکاف‌ها و تمایزات استفاده می‌شوند تا برای ارتقا برنامه‌ریزی شود.

برجسته‌ترین چارچوب ارزیابی فرآیند، مدل بلوغ قابلیت (CMM) نام دارد که توسط موسسه مهندسی نرم‌افزار (SEI) توسعه داده شد و در سال ۱۹۹۳ منتشر شد. سپس یکپارچه شد تا برای مهندسی نرم‌افزار در قالب مدل CMMI مورد استفاده قرار گیرد.

در زمینه مهندسی خانواده محصولات نرم‌افزاری، SEI یک چارچوب برای شیوه خط تولید نرم‌افزار منتشر کرد که در آن ۲۹ «عرصه عملی» در قالب سه دسته کلی تشخیص داده شده است:

- عرصه‌های عملی مهندسی نرم‌افزار برای به‌کارگیری فناوری‌های مناسب در ایجاد و رشد هر دو بخش دارایی‌های پایه و محصولات مورد نیاز هستند.
- عرصه‌های عملی مدیریت فنی لازم هستند تا برای مهندسی ایجاد و تکامل دارایی‌های پایه و محصولات استفاده شوند.
- عرصه‌های عملی مدیریت سازمانی برای هماهنگی تمامی فعالیت‌های خانواده محصولات نرم‌افزاری مورد نیاز هستند.

در روش «کاوش فنی خط تولید» (PLTP) که به سازمان SEI تعلق دارد، میزان برآورده کردن پذیرفتن یک رویکرد خانواده محصولات نرم‌افزاری مورد ارزیابی قرار می‌گیرد. در این روش از چارچوب عملی خط تولید نرم‌افزاری سازمان SEI به عنوان یک مدل مرجع در زمینه جمع‌آوری و تحلیل داده‌های پیرامون یک سازمان استفاده می‌شود. نتایج اجرای PLTP شامل مجموعه‌ای از یافته‌هاست که توصیف‌گر توانایی‌ها و چالش‌های پیش‌روی تلاش‌های خانواده محصولات و نیز مجموعه‌ای از توصیه‌هاست.

علاوه بر این مدل‌ها، مدل‌های اقتصادی پایه‌ای متعددی نیز وجود دارند که میزان موفقیت مهندسی خانواده محصولات نرم‌افزاری را ارزیابی می‌کنند. این روش‌ها سعی می‌کنند تا ارزش تجاری را از راه‌های مختلف ارزیابی کنند. به عنوان مثال Jan Bosch پیشنهاد دهنده هر دو مدل اولیه معماری‌های خانواده محصولات نرم‌افزاری و یک بازرسی اولیه در مورد ساختارهای سازمانی است.

از تمامی این رویکردها به روش‌های گوناگون در FEF استفاده می‌شود. به عنوان مثال از CMMI به عنوان بستری برای بعد فرآیند استفاده می‌شود. از چارچوب عملی خط تولید نرم‌افزار و مدل‌های معماری و اقتصادی نیز استفاده شود تا بررسی شود که آیا همه موارد مورد نیاز برای توسعه خانواده محصولات نرم‌افزاری در نظر گرفته شده است یا خیر. ضمناً از مدل‌های اقتصادی و معماری به عنوان یک الهام دهنده به ابعاد FEF که با این دو مولفه سر و کار دارند استفاده شده است. مدل FEF ساختار بیشتری را به این کل اضافه کرده است که باعث شده ارائه تصویر کامل‌تری از سازمان تضمین شود.

#### ۴-۱-۳- ارزیابی خانواده محصولات نرم‌افزاری

در این مقاله یک چارچوب ۴ بعدی برای ارزیابی مهندسی خانواده محصولات نرم‌افزاری مشخص می‌شود. این چهار بعد به دغدغه‌های مهندسی نرم‌افزار شامل کسب و کار، معماری، سازمان و فرآیند می‌پردازند. هدف این چارچوب این است که در سازمان‌های توسعه‌دهنده نرم‌افزار استفاده شود تا وضعیت مهندسی خانواده محصولات نرم‌افزاری آن‌ها و اولویت‌هایی برای ارتقا را مشخص کند. نتایج این ارزیابی می‌تواند در محک‌زدن سازمان‌ها، تهیه نقشه راه و توسعه طرح‌های بهبودی مورد استفاده قرار گیرد.

قضایای عمده‌ای که منجر به معرفی مهندسی خانواده محصولات نرم‌افزاری می‌شوند به این شرح هستند:

- افزایش قابلیت تولید
- بهبود پیش‌بینی پذیری
- کاهش زمان ورود به بازار
- افزایش کیفیت و قابلیت اطمینان

برای بهبود توانایی‌های سراسری مهندسی خانواده سیستم‌ها، تعدادی از پروژه‌های معماری سازمانی در قاره اروپا مورد ارزیابی قرار گرفته‌اند که از جمله آن‌ها می‌توان به پروژه‌های ESAPS if99005 (1999-2001)، CAFE if00004 (2001-2003) و FAMILIES if02009 (2003-2005) اشاره کرد.

در این بخش از مقاله یک چارچوب بهبود یافته به همراه یک توصیف مفصل‌تر از دغدغه‌های مختلف توسعه مرتبط با مهندسی خانواده محصولات نرم‌افزاری تهیه می‌شود. تمرکز این بخش از مقاله روی سامانه‌های



نهفته است. مهندسی خانواده محصولات نرم‌افزاری از توسعه سامانه‌های نهفته نشات گرفته است؛ جایی که خانواده محصولات قبلا وجود داشته‌اند. نرم‌افزار در سامانه‌های نهفته در آغاز برای بهبود انعطاف‌پذیری معرفی شد و سپس برای تعریف قابلیت‌های جدید مورد استفاده قرار گرفت. زمانی که حجم نرم‌افزار در حال گسترش بود، راهکارهای مهندسی خانواده محصولات باید روی نرم‌افزار نیز اعمال می‌شد. از آنجایی که نرم‌افزار نسبت به سایر دارایی‌های محصول رفتار متفاوتی دارد، در ابتدا چگونگی مواجهه با آن چندان واضح نبود. اما در این بخش از مقاله به تمامی دغدغه‌های عمده محصولات خانواده محصولات نرم‌افزاری پرداخته می‌شود راهکار معرفی شده برای انواع سامانه‌های نرم‌افزاری خالص نیز قابل استفاده است؛ هر چند که مثال‌های زیادی از این نوع محصولات مورد ارزیابی قرار نگرفته است.

ادامه این فصل از مقاله به این صورت سازمان‌دهی شده است: در بخش بعد، مدل BAPO (کسب و کار - معماری - فرآیند - سازمان) معرفی می‌شود. مدل BAPO به عنوان مبنایی برای چارچوب ارزیابی مورد استفاده قرار می‌گیرد. متعاقبا چهار بعد ارزیابی تشریح می‌شوند. سپس یک مثال ارزیابی آورده می‌شود و کارهای مرتبط به همراه نتیجه‌گیری تشریح خواهد شد.

#### ۴-۲- مدل BAPO

در پروژه معماری سازمانی فناوری اطلاعات ESAPS<sup>۱۳</sup> که مخفف عبارت «مهندسی معماری‌های نرم‌افزار، فرآیندها و چارچوب‌ها برای مهندسی خانواده سامانه‌ها» است، چهار دغدغه مستقل توسعه نرم‌افزار در قالب مدل شناسایی شد:

۱. کسب و کار (Business): چگونه می‌توان از محصولات خود سود مالی کسب کرد
۲. معماری (Architecture): جنبه‌های فنی ایجاد نرم‌افزار
۳. فرآیند (Process): نقش‌ها، مسئولیت‌ها و ارتباطات در توسعه نرم‌افزار
۴. سازمان (Organization): نگاشت حقیقی نقش‌ها و مسئولیت‌ها به ساختار سازمانی

چهار دغدغه عمده‌ای که در BAPO به آن پرداخته می‌شود در عین مستقل بودن همگی به هم مرتبط است زیرا اجرا کردن تغییرات در یک بعد منجر به بروز تغییرات در سایر ابعاد می‌شود. اصطلاح BAPO به یک ترتیب طبیعی برای پیمایش نیازمندی‌ها اشاره دارد. کسب و کار موثرترین معیار می‌باشد؛ بنابراین در نخستین گام به آن پرداخته می‌شود. معماری بازتاب‌دهنده نگرانی‌های تجاری در ساختار و قوانین نرم‌افزار است. فرآیند امکان ایجاد نرم‌افزار را فراهم می‌کند که بر پایه معماری و با توجه به قوانین کسب و کار صورت گرفته است. در نهایت سازمان باید میزبان فرآیند باشد که از طریق تخصیص دادن واحدها و افرادی که

<sup>۱۳</sup> Engineering Software Architectures, Processes and Platforms for system family Engineering

مسئولیت کسب و کار و معماری را دارند صورت می‌پذیرد. باید توجه داشت که سهولت و سرعت تغییر یک نیازمندی، با حرکت از سمت کسب و کار به معماری، فرآیند و سازمان کاهش می‌یابد.

ما از این جداسازی دغدغه‌ها استفاده می‌کنیم تا چهار مولفه چارچوب ارزیابی خانواده را فراهم کنیم. هر سازمان یک سطح ارزیابی جداگانه برای هر کدام از این دغدغه‌ها خواهد داشت. همبستگی بین دغدغه‌های BAPO زمانی روشن می‌شود که تاثیرات تغییرات مورد ارزیابی قرار گیرد. تغییرات در یک بعد به صورت مجازی همیشه نتایجی را در سایر ابعاد نیز خواهد داشت؛ با این وجود در موارد متعددی بهبود ابعاد به صورت جداگانه مورد بررسی قرار می‌گیرد. در نهایت همه جنبه‌های BAPO باید تحت نظر گرفته شود تا بهبود کلی حاصل شود. یک امتیاز پایین در یک بعد می‌تواند مانع کسب امتیاز بالا در سایر ریع‌ها شود. عملیاتی که برای بهبود نتایج حاصل از ارزیابی یک دغدغه شده است، ممکن است منجر به کاهش یا افزایش نتایج ارزیابی سایر ریع‌ها گردد. در نتیجه فعالیت‌های بهبود دهنده باید تمامی دغدغه‌های مدل BAPO را در نظر بگیرند.

از آنجایی که BAPO روی مهندسی نرم‌افزار اعمال می‌شود، فعالیت‌های آن روی فعالیت‌های مهندسی خانواده محصولات نرم‌افزاری نیز اعمال خواهد شد. در نتیجه FEF برای هر کدام از این دغدغه‌ها هم بعد متناظر خود را دارد که شامل جنبه‌ها و سطوح است. برای هر بعد میزان پیوستگی مهندسی خانواده محصولات نرم‌افزاری ارزیابی می‌شود. برای هر بعد، راه‌های مشخص متعددی وجود دارد تا مهندسی محصولات نرم‌افزاری را انجام داد. مشابه فلسفه CMMI، ارزیابی FEF سر و کاری با راه‌های خاص انجام یک فعالیت مشخص، مدل‌سازی، ساختاردهی، مسئولیت یا کار ویژه‌ای ندارد.

پروژه‌های ESAPS، CAFÉ و FAMILIES یک حجم عمده‌ای از این متدها، ابزارها و تکنیک‌ها را فراهم کرده‌اند. این‌ها بهترین تجارب هستند و می‌توانند در مهندسی سامانه‌های نرم‌افزاری اعمال شوند. این‌ها یک بینش مناسب در مورد مواردی که در ریع‌های مختلف BAPO مورد نیاز است فراهم می‌کنند. با این وجود هیچ‌کدام از آن‌ها اجباری نیستند. اغلب روش‌ها، ابزارها و تکنیک‌های خاص با گذر زمان تغییر می‌کنند.

برای اجرای چهار بعد BAPO در مهندسی خانواده نرم‌افزارها، جنبه‌های زیر باید در نظر گرفته شود:

- **B:** کسب و کار باید علاوه بر دغدغه‌های خود، با ارتباطات تجاری بین مهندسی دامنه و برنامه و هزینه‌ها، درآمدها، ارزش بازار و برنامه‌ریزی تغییرپذیری نیز سر و کار داشته باشد.
- **A:** معماری به دو بخش دامنه و برنامه تقسیم می‌شود که از طریق تغییرپذیری به یکدیگر مرتبط هستند. دغدغه‌های مهم معماری با ساز و کارهای صحیح برای تغییرپذیری سر و کار دارند و این که

چگونه معماری خانواده محصولات نرم‌افزاری (مهندسی دامنه) روی معماری برنامه‌های متناظر (مهندسی برنامه) و جهت خلاف آن تاثیر می‌گذارد.

- **P:** فرآیندها را می‌توان به دسته فرآیندهای دامنه، برنامه و همکاری و هماهنگی تقسیم کرد. هر کدام از آن‌ها را می‌توان با استفاده از یک مدل بلوغ مانند CMMI ارزیابی کرد.
- **O:** سازمان دارای ساختارها و مسئولیت‌هایی برای مهندسی دامنه و/یا برنامه و نیز نقش‌های همکاری و هماهنگی است. به طور خاص، سازمان مسئولیت‌ها را بین چارچوب (مهندسی دامنه)، برنامه‌های متناظر (مهندسی برنامه) و همکاری و هماهنگی توزیع می‌کند و اهمیت نسبی بین آن‌ها را مشخص می‌کند.

#### ۴-۳- چارچوب FEF

همان‌گونه که در بخش قبل ذکر شد، چارچوب FEF بر اساس مدل BAPO ساخته شده است و در آن هر کدام از چهار دغدغه مدل BAPO به صورت مجزا مورد ارزیابی قرار می‌گیرد بدین صورت که برای هر کدام از این ربع‌ها، یک مجموعه‌ای از جنبه‌های ارزیابی مشخص می‌شود که تشکیل دهنده معیارهای اساسی تاثیرگذار روی ارزیابی است. سپس مقادیر به دست آمده از هر ربع در قالب یکی از پنج سطح کیفی، گزارش می‌شوند.

«چارچوب ارزیابی خانواده»<sup>۱۴</sup> (FEF) یکی از نتایج متفق پروژه FAMILIES محسوب می‌شود. در این پروژه نتایج تجربیات با خانواده محصولات نرم‌افزاری ESAPS، CAFÉ و FAMILIES در قالب یک چارچوب ادغام شده است. هدف از FEF، ارزیابی کارآیی مهندسی خانواده محصولات نرم‌افزاری در بخش‌های بزرگتر یا کوچکتر سازمان‌ها شامل بخش‌های تجاری، واحدها، و حتی کل سازمان است. این ارزیابی بر پایه بهترین تجارب و سایر تجربیات شرکت‌های عضو در این پروژه‌ها بوده است.

دلایل متعددی برای فواید استفاده از چارچوب ارزیابی FEF وجود دارد. به عنوان مثال از آن می‌توان برای یافتن تخمینی از این که یک واحد سازمانی چگونه مهندسی خانواده محصولات نرم‌افزاری را انجام می‌دهد استفاده کرد. از طرف دیگر، از آن می‌توان به عنوان یک ابزار محک استفاده کرد تا قابلیت‌های مهندسی خانواده محصولات نرم‌افزاری یک سازمان را با سایر سازمان‌ها مقایسه کرد. ضمناً از آن می‌توان به عنوان یک ابزار تصمیم‌گیری استفاده کرد که به کمک آن در مورد کاری که برای شروع یا بهبود مهندسی خانواده محصولات نرم‌افزاری در یک سازمان خاص باید انجام شود، اطلاعات کسب کرد.

در ادامه این سند یک معرفی کلی در مورد FEF ارائه می‌شود. چهار بعد مختلف در این چارچوب وجود دارند که هر کدام می‌تواند به صورت مستقل مورد ارزیابی قرار گیرد. درون هر بعد، مجموعه‌ای از جنبه‌های ارزیابی و نیز مجموعه‌ای از سطوح وجود دارند. در این سند هر کدام از این سطوح در قالب روش‌هایی که جنبه‌ها با آن سر و کار دارند توصیف می‌شود که در ادامه، به صورت مفصل‌تر در مورد این جنبه‌ها بحث خواهد شد.

#### ۴-۴- بعد کسب و کار (Business)

بعد کسب و کار در حالت کلی با چگونگی کسب سود از محصولات سروکار دارد. در انجام این کار، بعد کسب و کار چندین تکنیک دارد که با آن‌ها روی فرایند ایجاد تأثیر می‌گذارد و بازار و فروش محصولات را بهبود می‌بخشد. خیلی از مسائل کسب و کار در همه سازمان‌هایی که نرم‌افزار می‌سازند برقرار هستند و در اینجا در مورد آن‌ها صحبت نخواهیم کرد. البته در مهندسی خط تولید، بعد کسب و کار افزودن بر موارد معمول، باید با ارتباطات کسب و کار بین مهندسی دامنه و برنامه هم سروکار داشته باشد، که این‌ها شامل تصمیمات سرمایه‌گذاری، اندازه‌گیری هزینه و سود این فعالیت‌ها و نحوه استفاده از فروش محصولات برای پرداخت هزینه‌های فعالیت‌های مختلف می‌شوند. در نهایت، بعد کسب و کار با variability مدیریت شده سروکار دارد. بعد کسب و کار، هزینه و سود میزان variability در محصولات را اندازه‌گیری می‌کند. همچنین از آن برای مشخص کردن قیمت‌ها و راهبردهای بازاری برای محصولات خط تولید، استفاده می‌کند.

جنبه‌های زیر در بعد کسب و کار مهندسی خط تولید نرم‌افزار، نقش ایفا می‌کنند:

- تأثیر فروش، بازار و مدیریت محصولات

چقدر بازار، فروش و مدیریت محصولات در خط تولید تأثیر می‌گذارند و از آن تأثیر می‌پذیرند.

- بودجه و سرمایه‌گذاری

بودجه چقدر تحت تأثیر مهندسی خط تولید نرم‌افزار قرار می‌گیرد؟

- دید و اهداف کسب و کار

سازمان تا چه اندازه برای آینده شامل مهندسی خط تولید نرم‌افزار هدف‌گیری می‌کند.

- برنامه‌ریزی راهبردی

سازمان تا چه اندازه ایجاد و مسائل کسب و کار خط تولید را برنامه‌ریزی می‌کند.

سطح‌های زیر برای بعد کسب و کار مهندسی خط تولید نرم‌افزار شناخته شده‌اند.

۱. Project based
۲. Aware
۳. Managed
۴. Measured
۵. Optimised

در بخش‌های بعدی، هرکدام از این سطوح‌های با جزئیات بیشتری شرح داده شده‌اند.

#### ۴-۴-۱- سطح اول: Project based

این سطح، سطح اولیه است. تجارت سازمان برپایه مهندسی تک‌سیستم ترتیب داده شده است. نتایج مهندسی دامنه و variability در سطح کسب و کار قابل مشاهده نیستند. هیچ کدام از جنبه‌ها مهندسی خط تولید نرم‌افزار را پوشش نمی‌دهند. با توجه به دغدغه‌های کسب و کار، وضعیت‌های نوعی زیر را می‌بینیم:

- تأثیر فروش، بازار و مدیریت محصولات

نه تنها هیچ نشانه‌ای از مهندسی خط تولید نرم‌افزار نیست (یا مقدار بسیار اندکی است)، حتی درک زیادی از آن هم وجود ندارد. سیستم‌ها براساس یک پایه تک‌سیستمی، برنامه‌ریزی، ایجاد و فروخته می‌شوند.

- بودجه و سرمایه‌گذاری

هیچ بودجه مشخصی برای مهندسی دامنه وجود ندارد. به جای آن، بودجه‌ها برای تک‌سیستم‌ها تخصیص داده می‌شوند.

- دید و اهداف کسب و کار

دید و اهداف کسب و کار سازمان، هیچ اشاره‌ای به وجود مهندسی خط تولید نرم‌افزار نمی‌کنند.

- برنامه‌ریزی راهبردی

برنامه‌ریزی کسب و کار، ارتباط بین سیستم‌ها را در نظر نمی‌گیرد.

#### ۴-۴-۲ - سطح دوم: Aware

در این سطح، تجارت سازمان از مزایای مهندسی خط تولید نرم‌افزار برای سازمان آگاه است. به همین منظور زمینه‌ای مهیا می‌کند که بتوان مهندسی خط تولید نرم‌افزار را انجام داد. اگرچه هنوز یک مدیریت واضح مهندسی خط تولید نرم‌افزار وجود ندارد.

این سطح، وضعیت‌های معمول زیر را نشان می‌دهد:

- تأثیر فروش، بازار و مدیریت محصولات

توسط نیروی فروش، بازاریابی و مدیریت محصولات، یک آگاهی از فرصت‌هایی که مهندسی خط تولید نرم‌افزار در بازاریابی، فروش و برنامه‌ریزی محصولات ایجاد می‌کند، به وجود می‌آید. انتظار می‌رود که variability مدیریت شده به variability بیشتر در فروش سیستم‌ها منجر می‌شود و هزینه تولید پایین‌تر می‌آید. تنها همین مسأله پشتیبانی variant‌های بیشتر، به عنوان یک نفع اضافی برای کاربر دیده می‌شود. البته راه‌برد مشخصی برای استفاده از خط تولید نرم‌افزار در بازاریابی، فروش و برنامه‌ریزی محصولات وجود ندارد.

- بودجه و سرمایه‌گذاری

تجارت سازمان روی فعالیت‌های مهندسی دامنه به منظور پشتیبانی از یک منبع برای دارایی‌های قابل استفاده دوباره، سرمایه‌گذاری می‌کند. به منظور تشویق برای استفاده از نتایج مهندسی دامنه، بودجه‌هایی در نظر گرفته می‌شود.

- دید و اهداف کسب و کار

یک تعهد از طرف مدیریت بالا برای انجام مهندسی خط تولید نرم‌افزار وجود دارد. اگرچه دید واضحی نسبت به کاربرد آن برای سازمان وجود ندارد.

- برنامه‌ریزی راه‌بردی

برنامه‌ریزی هنوز به شکل تک‌سیستم انجام می‌شود. اگرچه نتایج مهندسی دامنه به عنوان یک فرصت در نقشه‌های راه، در نظر گرفته شده است.

#### ۴-۳-۳ - سطح سوم: Managed

در این سطح، مهندسی خط تولید نرم‌افزار، بخشی از راه‌برد کسب و کار است و کنترل اجرای فعالیت‌های مرتبط، مزایا و معایب آن را برعهده می‌گیرد.

- تأثیر فروش، بازار و مدیریت محصولات

بازگشت سرمایه (ROI)، بازاریابی، فروش و ایجاد محصولات خط تولید نرم‌افزار را جلو می‌برد. بازاریابی نشان می‌دهد که کاربران به داشتن مقدار زیاد variability با هزینه کم، ارزش می‌دهند.

- بودجه و سرمایه‌گذاری

مهندسی خط تولید نرم‌افزار، تصمیمات سرمایه‌گذاری را تحت تأثیر می‌گذارد. یک بودجه تعریف شده برای فعالیت‌های مهندسی دامنه و برنامه وجود دارد. یک مکانیزم سازمان یافته برای ایجاد بودجه برای مهندسی دامنه توسط فروش سیستم‌های تولید شده با مهندسی برنامه، وجود دارد. همچنین یک آگاهی از هزینه‌ها و سودهای variability و اینکه چگونه بازگشت سرمایه ایجاد می‌کند، وجود دارد.

- دید و اهداف کسب و کار

مدیریت بالا، قویاً از مدیریت مهندسی خط تولید نرم‌افزار پشتیبانی می‌کند. دید و اهداف کسب و کار سازمان در خط تولید نرم‌افزار، ارزش آن برای سازمان و تکامل آن مشارکت می‌کند. راه‌برد مهندسی خط تولید نرم‌افزار، برای سازمان قابل مشاهده است.

- برنامه‌ریزی راه‌بردی

طرح‌ها و نقشه‌های راه جدا برای مهندسی دامنه و برنامه وجود دارد. این طرح‌ها به هم مرتبط هستند و اشتراکات برنامه‌ها، پایه طرح مهندسی دامنه را ایجاد می‌کنند.

#### ۴-۴-۴ - سطح چهارم: Measured

در این سطح، تجارت سازمان، مهندسی خط تولید نرم‌افزار را به منظور بهبود راه‌برد، اندازه‌گیری می‌کند.

- تأثیر فروش، بازار و مدیریت محصولات

هزینه‌ها، سود و بازگشت سرمایه محصولات خط تولید نرم‌افزار و variability مدیریت شده به منظور مشخص کردن راه‌برد بازاریابی و فروش، اندازه‌گیری می‌شوند. علاوه بر این، راه‌برد مدیریت محصولات با بازگشت سرمایه اندازه‌گیری شده راهنمایی می‌شوند.

- بودجه و سرمایه‌گذاری

هزینه‌ها و صرفه‌جویی‌های استفاده دوباره و variability و مهندسی خط تولید نرم‌افزار اندازه‌گیری شده و در بودجه‌ها در نظر گرفته می‌شوند.

- دید و اهداف کسب و کار

مدیریت بالا، تأثیرات ایجاد خط تولید نرم‌افزار را اندازه‌گیری می‌کند. اهداف کسب و کار، در خط تولید نرم‌افزار، ارزش آن برای سازمان و تکامل آن، تأثیر می‌گذارد. مزایای خط تولید نرم‌افزار، در دید و اهداف کسب و کار، خود را نشان می‌دهد. معایب شناسایی می‌شود و اقدام‌ها لازم برنامه‌ریزی می‌شود تا تأثیرات آن‌ها را از بین ببرد. راه‌برد مدیریت خط تولید نرم‌افزار در بیرون از سازمان (مشتری‌ها، سرمایه‌گذارها و ...) دیده می‌شود.

- برنامه‌ریزی راه‌بردی

طرح‌ها و نقشه‌های راه طوری جهت‌دهی می‌شوند که بهترین ارزش کسب و کار از مهندسی خط تولید نرم‌افزار، دریافت شود.

#### ۴-۴-۵- سطح پنجم: Optimised

در این سطح، راه‌برد کسب و کار شامل بهینه‌سازی مهندسی خط تولید نرم‌افزار می‌شود.

- تأثیر فروش، بازار و مدیریت محصولات

بازاریابی، فروش و ایجاد، درکی از مهندسی خط تولید نرم‌افزار داشته و از ارزش آن در راه‌برد استفاده می‌کند. بازاریابی و فروش از هزینه‌ها، سود و بازگشت سرمایه مهندسی خط تولید نرم‌افزار، استفاده می‌کند. این هزینه‌ها و سودها، برای بهبود راه‌برد کسب و کار هم استفاده می‌شود.

- بودجه و سرمایه‌گذاری

یک تجمیع دقیق بین بودجه و سرمایه‌گذاری با پیش‌بینی فروش، هزینه‌ها و صرفه‌جویی‌های محصولات خط تولید نرم‌افزار وجود دارد.

- دید و اهداف کسب و کار

دید و اهداف کسب و کار تحت تأثیر خط تولید نرم‌افزار با یک پایه کاملاً درک شده، قرار می‌گیرد.



- برنامه‌ریزی راه‌بردی

طرح‌ها و نقشه‌های راه به شکل راه‌بردی استفاده می‌شود تا بهترین ارزش کسب و کار از مهندسی خط تولید نرم‌افزار به دست آید.

#### ۴-۴-۶- خلاصه

سطح‌های ارزیابی برای بعد کسب و کار عبارتند از: Measured, Managed, Aware, Project based و Optimised. در سطح اولیه، هیچ دخالت واقعی کسب و کار در مهندسی خط تولید نرم‌افزار وجود ندارد. بعد کسب و کار با یک سازمان که بر مبنای تک‌سیستم کار می‌کند، سروکار دارد و با همه پروژه‌ها به شکل مشابه توسط مدیریت رفتار می‌شود. بعد کسب و کار FEF با جنبه‌های زیر سروکار دارد

- تأثیر فروش، بازار و مدیریت محصولات

با شروع از یک وضعیت ناآگاه، نیروی مدیریت محصولات، بازاریابی و فروش از احتمالات استفاده از variability مدیریت شده، آگاه می‌شود. در سطح بالاتر، بازاریابی و فروش، رشد کرده و از مدیریت variability به شکلی استفاده می‌کند که برنامه‌ریزی محصولات از فروش مورد انتظار، پشتیبانی کرده و بازاریابی هم از variant‌های موجود در خانواده محصولات پشتیبانی می‌کند.

- بودجه و سرمایه‌گذاری

در ابتدا هیچ بودجه یا سرمایه‌گذاری خاصی برای مهندسی خط تولید نرم‌افزار وجود ندارد. در سطح بعد، مدیریت از مزایای احتمالی مهندسی خط تولید نرم‌افزار آگاه بوده ولی هنوز تصمیمی مبنی بر اینکه چگونه به بهترین نحو ممکن از این مسأله استفاده کند، نگرفته است. سرمایه‌گذاری‌های اولیه برای مهندسی دامنه از طرف مدیریت بالا، پشتیبانی می‌شود. در سطح بالاتر، بودجه و سرمایه‌گذاری مهندسی دامنه، تخصصی‌تر شده و از موقعیت مهندسی دامنه در سازمان، پشتیبانی می‌کند. مهندسی دامنه، بودجه خود را از طریق یک مدل کسب و کار داخلی مناسب، به دست می‌آورد.

- دید و اهداف کسب و کار

در ابتدا فقط کسانی که کار مهندسی خط تولید نرم‌افزار را انجام می‌دهند، دید دارند. سپس مدیریت آگاه شده و variability مدیریت شده را در دید و اهداف آینده، قرار می‌دهند.

- برنامه‌ریزی راه‌بردی

از یک مرحله اولیه، که مهندسی خط تولید نرم‌افزار در طرح‌ها قابل مشاهده نیست، شروع شده و در نهایت تبدیل به یک نیروی پیش‌راننده مهم در برنامه‌ریزی‌ها می‌شود.

#### ۴-۵- بُعد معماری

بعد معماری ناظر به جنبه‌ها و روش‌های فنی ساخت نرم‌افزار است. البته «معماری» در این جا گسترده‌تر از مفهوم سنتی معماری است، چرا که با مواردی مثل نیازمندی‌ها، مدل‌سازی دامنه، طراحی جزئی<sup>۱۵</sup>، و آزمون نیز سروکار دارد. با این حال، آن مفهوم سنتی، در این جا نیز مهم‌ترین موضوع است و تحقق<sup>۱۶</sup> محصول را در خانواده‌ی محصولات، از دید فنی تبیین خواهد کرد. به علاوه معماری در خط تولید به دو بخش معماری دامنه و معماری برنامه‌ی کاربردی<sup>۱۷</sup> تقسیم می‌شود، و این دو بخش از طریق مفهوم تغییرپذیری<sup>۱۸</sup> مرتبط خواهند شد. ارزیابی در بُعد معماری نیز عمدتاً در حوزه‌ی همین ارتباط انجام می‌شود. به بیان دیگر، باید مشخص شود که تغییرپذیری چه‌گونه در معماری مدل شده است. مهم‌ترین دغدغه‌ها در بُعد معماری معطوف به سازوکارهای صحیح تغییرپذیری، و طبعاً ارتباط معماری خانواده‌ی محصولات (مربوط به مهندسی دامنه) و معماری خود آن‌ها (مربوط به مهندسی برنامه‌ی کاربردی) است.

موارد زیر در بُعد معماری قابل توجه هستند:

- میزان استفاده‌ی مجدد<sup>۱۹</sup> از دارایی‌ها<sup>۲۰</sup>: چه قدر از دارایی‌های مربوط به دامنه در محصولات استفاده شده است؟
- معماری خانواده‌ی محصولات: معماری محصولات (برنامه‌های کاربردی) چه قدر بر معماری خانواده‌ی محصولات (معماری مرجع) مبتنی است؟
- مدیریت تغییرپذیری: کاربرد آشکار نقاط تغییرپذیری<sup>۲۱</sup> و سازوکارهای مرتبط.

#### سطوح

سطوح زیر در بعد معماری خانواده‌ی محصولات قابل تعریف است:

۱. توسعه‌ی مستقل محصولات

<sup>15</sup> detailed design  
<sup>16</sup> realisation  
<sup>17</sup> application  
<sup>18</sup> variability  
<sup>19</sup> reuse  
<sup>20</sup> asset  
<sup>21</sup> variation point

۲. زیرساخت مستقل از دامنه، ولی استاندارد شده

۳. پلتفرم نرم‌افزاری

۴. محصولات گوناگون قابل اشتقاق<sup>۲۲</sup>

۵. اشتقاق خودکار محصولات

هر یک از این سطوح را در ادامه به طور مفصل شرح خواهیم کرد.

#### ۴-۵-۱- سطح ۱: توسعه‌ی مستقل محصولات

**توضیح کلی:** این سطح ابتدایی‌ترین سطح است. معماری تنها برای تک‌سامانه‌ها وجود دارد. استفاده‌ی مجدد در مورد معماری قابل مشاهده نیست. هیچ جنبه‌ای مرتبط با مهندسی خط تولید وجود ندارد.

- **میزان استفاده‌ی مجدد از دارایی‌ها:** استفاده‌ی مجدد وجود ندارد، یا به شکل غیرنظام‌مند انجام می‌شود.

- **معماری خانواده‌ی محصولات:** چنین معماری‌ای وجود ندارد.

- **مدیریت تغییرپذیری:** تغییرپذیری‌ای وجود ندارد.

#### ۴-۵-۲- سطح ۲: زیرساخت مستقل از دامنه، ولی استاندارد شده

**توضیح کلی:** در این سطح، استفاده‌ی مجدد در حوزه‌ی زیرساخت طرف سوم<sup>۲۳</sup> انجام می‌شود. یک زیرساخت مشترک (نظیر میان‌افزار<sup>۲۴</sup> یا COTS<sup>۲۵</sup>) نیز تعریف خواهد شد. با این حال استفاده‌ی مجدد دارایی‌ها، به شکل رسمی دیده نمی‌شود.

- **میزان استفاده‌ی مجدد از دارایی‌ها:** یک زیرساخت مشترک مربوط به طرف سوم، تعریف می‌شود و به کار می‌رود. استفاده‌ی مجدد تنها به شکل سردستی انجام می‌شود، و عمدتاً متکی بر مجموعه‌ای از محصولات مربوط به طرف سوم است.

- **معماری خانواده‌ی محصولات:** این معماری از دل زیرساخت مربوط به طرف سوم استخراج می‌شود، و تنها اطمینان حاصل می‌کند که از آن زیرساخت استفاده شود.

<sup>22</sup> derivable

<sup>23</sup> third-party

<sup>24</sup> middlewear

<sup>25</sup> Commercial Off-The-Shelf

- **مدیریت تغییرپذیری:** تنها تغییراتی که به زیرساخت مربوط به طرف سوم ارتباط داشته باشد، مدیریت خواهد شد. تصمیم در مورد سایر تغییرات به معماری برنامه‌ی کاربردی محول می‌شود.

#### ۴-۵-۳- سطح ۳: پلتفرم نرم‌افزاری

**توضیح کلی:** در این سطح، مشترکات<sup>۲۶</sup> دامنه در قالب یک پلتفرم نرم‌افزاری ثبت، و پیاده‌سازی می‌شوند. یک معماری مرجع، نحوه‌ی استفاده از یک پلتفرم مشترک را برای تمام برنامه‌های کاربردی مشخص خواهد کرد. این پلتفرم برای محصولات گوناگون استفاده می‌شود، و قابل پیکربندی است. با این حال، تغییرپذیری (برای اشتقاق محصولات) پشتیبانی نمی‌شود.

- **میزان استفاده‌ی مجدد از دارایی‌ها:** پلتفرم مشترک را به صورت مجموعه‌ای از دارایی‌های مشترک تعریف می‌کنیم که در یک مخزن<sup>۲۷</sup> مربوط به دامنه نگه‌داری می‌شوند. استفاده‌ی مجدد به این پلتفرم منحصر است، و محدودیت‌های معماری نیز حدودی را بر آن اعمال می‌کند.
- **معماری خانواده‌ی محصولات:** معماری مرجع، شامل قواعد مشترکی است، و نحوه‌ی استفاده از پلتفرم مشترک را مشخص می‌کند. این قواعد، شامل مواردی مربوط به کیفیت نیز می‌شود. این معماری مرجع توسط برنامه‌های کاربردی به کار می‌رود.
- **مدیریت تغییرپذیری:** معماری مرجع تعیین می‌کند که استفاده از چه ترکیبی (پیکربندی‌ای) از دارایی‌های دامنه در برنامه‌های کاربردی مجاز است. نقاط تغییرپذیری نیز، که گونه‌های<sup>۲۸</sup> ویژه‌ی هر برنامه از آن جا می‌توانند متصل شوند، در معماری مرجع تعریف می‌شود.

#### ۴-۵-۴- سطح ۴: محصولات گوناگون قابل اشتقاق

**توضیح کلی:** در این سطح، مشترکات و تغییرپذیری‌های دامنه ثبت، و یک معماری مرجع برای یک خانواده‌ی کامل محصولات نرم‌افزاری تعریف می‌شود. دارایی‌های دامنه از محصولات قابل تغییر پشتیبانی می‌کنند. مدیریت تغییرپذیری نیز به طور صریح در معماری مرجع لحاظ می‌شود.

- **میزان استفاده‌ی مجدد از دارایی‌ها:** استفاده‌ی مجدد، به شکل نظام‌مند و مدیریت‌شده، با اتکا بر مخزنی از دارایی‌ها انجام می‌شود.

<sup>26</sup> commonality

<sup>27</sup> repository

<sup>28</sup> variant

- **معماری خانوادگی محصولات:** یک معماری مرجع به شکل واضح و صریح مشخص می‌کند که معماری برنامه‌های کاربردی در چه موضعی قابلیت تغییر دارد. به علاوه موارد مربوط به کیفیت نیز در این معماری مرجع لحاظ می‌شود.

- **مدیریت تغییرپذیری:** معماری یادشده در بالا مشخص می‌کند که چه پیکربندی‌ای برای معماری برنامه‌های کاربردی مجاز است. این معماری نقاط تغییرپذیری را مشخص، و تغییرات را به این نقاط محدود می‌کند. به علاوه قواعدی را تعریف می‌کند که گونه‌های ویژه‌ی برنامه‌های کاربردی باید آن‌ها را رعایت کنند.

#### ۴-۵-۵- سطح ۵: اشتقاق خودکار محصولات

**توضیح کلی:** در این سطح، مهندسی دامنه کاملاً مسلط است و نقش مهندسی برنامه‌های کاربردی حاشیه‌ای است. محصولات می‌توانند به شکل خودکار از دامنه مشتق شوند، و نیازی به توسعه‌ی ویژه‌ی هر محصول نیست. معماری مرجع نیز پیکربندی خودکار محصولات را ممکن می‌کند.

- **میزان استفاده‌ی مجدد از دارایی‌ها:** استفاده‌ی مجدد به شکل نظام‌مند، با اتکا به مخزن دارایی‌ها انجام می‌شود. این دارایی‌ها به شکل صریح مشتمل بر مواردی از تغییرپذیری‌اند، به علاوه‌ی سازوکاری برای اشتقاق خودکار محصولات.

- **معماری خانوادگی محصولات:** معماری مرجع، معماری برنامه‌های کاربردی را به طور کامل مشخص می‌کند. ضمن این که از پیکربندی خودکار، با هدف اشتقاق محصولات گوناگون پشتیبانی به عمل می‌آورد. موارد کیفی نیز با استفاده‌ی مدیریت‌شده از نقاط تغییر پیش‌بینی‌شده، لحاظ می‌شوند.

- **مدیریت تغییرپذیری:** مدیریت تغییرپذیری کاملاً با معماری یک‌پارچه شده است. تغییرپذیری به کمک مدل‌ها و/یا با استفاده از زبان‌های معنایی یا نحوی (که به شکل محلی استاندارد هستند) توصیف می‌شود. گونه‌ها نیز به شکل خودکار مشتق می‌شوند.

#### ۴-۵-۶- خلاصه

سطوح ارزیابی برای بُعد معماری عبارت بودند از: توسعه‌ی مستقل محصولات، زیرساخت مستقل از دامنه ولی استانداردشده، پلتفرم نرم‌افزاری، محصولات گوناگون قابل اشتقاق، و اشتقاق خودکار محصولات. در سطح ابتدایی، معماری دامنه وجود ندارد، و هر محصول معماری مربوط به خود را دارد. ضمن این که استفاده‌ی مجدد نیز غیرنظام‌مند و سردستی است.

- **میزان استفاده‌ی مجدد از دارایی‌ها:** از سطح ابتدایی استفاده‌ی مجدد غیرنظام‌مند، این میزان از طریق یک زیرساخت مشترک برای بخش‌های بزرگ‌تر معماری (که استفاده‌ی مجدد می‌شود) رشد می‌کند، و در نهایت به سطحی می‌رسد که استفاده‌ی مجدد از طریق نقاط تغییرپذیری مدیریت می‌شود.
- **معماری خانواده‌ی محصولات:** از سطح ابتدایی نبود هیچ‌گونه معماری در دامنه، به سمت پدیدآمدن یک معماری مرجع می‌رویم که در بالاترین سطح، همه‌ی برنامه‌های کاربردی را مدیریت می‌کند و آزادی کمی برای معماری هر کدام باقی می‌گذارد.
- **مدیریت تغییرپذیری:** از سطح ابتدایی عدم مدیریت تغییرپذیری، به تدریج به سطحی می‌رویم که تغییرپذیری کاملاً با معماری یک‌پارچه می‌شود و گونه‌ها به شکل خودکار قابل اشتقاق خواهند بود.

#### ۴-۶- بعد فرایند

فرایند با نقش‌ها، مسئولیت‌ها و ارتباطات در ایجاد نرم‌افزار سروکار دارد. فرایند به نحوه انجام فعالیت‌ها با هدف ایجاد نرم‌افزار می‌پردازد. برای مهندسی خط تولید نرم‌افزار فرایندها را می‌توان به سه بخش دامنه، برنامه و همکاری تقسیم کرد. فرایندهای مهندسی دامنه و برنامه، فرایندهای ایجاد هستند و CMMI را می‌توان به هر کدام به صورت جداگانه اعمال کرد. از آنجایی که فرایندهای مهندسی برنامه با مهندسی دامنه و دیگر فرایندهای مهندسی باید هماهنگ باشند، فرایندهای همکاری دیگری هم باید موجود باشند.

جنبه‌های زیر هر کدام نقشی را در بعد فرایند در مهندسی خط تولید نرم‌افزار بازی می‌کنند:

- مهندسی دامنه: فرایندهایی که کار مهندسی دامنه را انجام می‌دهند.
- مهندسی برنامه: فرایندهایی که کار مهندسی برنامه را انجام می‌دهند.
- همکاری: فرایندهایی که فعالیت‌های همکاری و هماهنگ‌سازی بین مهندسی برنامه و دامنه را انجام می‌دهند.

#### سطح‌ها

سطح‌های بعد فرایند در مهندسی خط تولید نرم‌افزار برپایه سطح‌های CMMI قرار گرفته‌اند. سطح‌های مهندسی خط تولید نرم‌افزار حاوی گسترش‌هایی بر فعالیت‌های CMMI در همان سطح هستند. برای اطلاعات بیشتر در مورد بعد فرایند به محصولات وظیفه 2.2 [5] مراجعه کنید. فرایندها بین مهندسی دامنه

و برنامه جدا شده‌اند. فرایندهای بیشتری برای همکاری بین مهندسی دامنه و برنامه لازم است. برپایه CMMI سطوح زیر برای بعد فرایند در مهندسی خط تولید نرم‌افزار تشخیصی به رسمیت شناخته شده‌اند.

- ۱. Initial
- ۲. Managed
- ۳. Defined
- ۴. Quantitatively managed
- ۵. Optimising

در بخش‌های بعدی هر کدام از این سطوح با جزئیات بیشتر بررسی شده‌اند.

#### ۴-۶-۱- سطح اول: Initial

این سطح، سطح اولیه است، مهندسی دامنه و برنامه و فرایندهای همکاری در سطح 1 CMMI انجام می‌شوند.

- مهندسی دامنه: در صورت وجود، مهندسی دامنه در سطح 1 CMMI انجام می‌گیرد.
- مهندسی برنامه: مهندسی برنامه در سطح 1 CMMI انجام می‌پذیرد.
- همکاری: در صورت وجود، همکاری در سطح 1 CMMI انجام می‌شود.

#### ۴-۶-۲- سطح دوم: Managed

در این سطح، مدیریت پروژه در جای خود قرار می‌گیرد. برای مهندسی خط تولید نرم‌افزار، پروژه‌های مهندسی برنامه و دامنه همگام‌سازی می‌شوند.

- مهندسی دامنه
- مهندسی دامنه در سطح 2 CMMI انجام می‌شود. البته برای محیط‌های فرایند زیر گسترش‌هایی لازم است:
  - مدیریت نیازمندی‌ها (RM): نیازمندی‌های خط تولید نرم‌افزار را مدیریت کنید. قابلیت ردیابی بین variation point و variantها را نگهداری و برقرار کنید.
  - طرح‌ریزی پروژه (PP): موارد variability را تعریف کنید.
  - نظارت و کنترل پروژه (PMC)
  - اندازه‌گیری و تحلیل (MA): نمای خط تولید کلی را به حساب بیاورید.
  - مدیریت تنظیمات (CM): باید به معماری پایه و انتشار assetها توجه کرد.
- مهندسی برنامه
- مهندسی برنامه در سطح 2 CMMI انجام می‌شود. برای محیط‌های فرایند زیر گسترش‌هایی لازم است:
  - مدیریت نیازمندی‌ها (RM): مدیریت نیازمندی‌ها در هر دو حالت نیازمندی‌های assetهای استفاده شده و نیازمندی‌های خاص برنامه باید انجام شود.

- طرح‌ریزی پروژه (PP): از assetهای دامنه استفاده کرده و variabilityهای آنها را با مقدار مناسب جایگزین کنید. ریسک وابستگی به مهندسی دامنه را تحلیل کنید. مهندسی دامنه را به عنوان یک ذی‌نفع برای ایجاد assetهای دامنه قابل استفاده دوباره در نظر بگیرید. همچنین تأثیر مهندسی دامنه بر حوزه یک پروژه را هم در نظر بگیرید.
- نظارت و کنترل پروژه (PMC): بر استفاده از assetها نظارت داشته باشید.
- اندازه‌گیری و تحلیل (MA): استفاده از assetهای مشترک توسط فعالیت‌های مهندسی برنامه را اندازه‌گیری کنید.
- مدیریت تنظیمات (asset): CMهای قابل استفاده دوباره یک پایه برای مشخص کردن و تنظیم آیتم‌ها مهیا می‌کنند.

• همکاری

همکاری در سطح CMMI 2 انجام می‌شود. برای محیط‌های فرایند زیر گسترش‌هایی لازم است:

- مدیریت نیازمندی‌ها (RM): قابلیت ردیابی دوطرفه بین خط تولید نرم‌افزار و نیازمندی‌های برنامه برقرار و نگهداری کنید. از آن برای مشخص کردن ناسازگاری‌ها استفاده کنید.
- طرح‌ریزی پروژه (PP): طول عمر assetها بیشتر از پروژه‌ها است. بین مهندسی برنامه و دامنه همگام‌سازی انجام دهید. اشتراکات بین مهندسی دامنه و برنامه را نظارت کنید.
- نظارت و کنترل پروژه (PMC): نقاط همگام‌سازی بین مهندسی دامنه و برنامه را نظارت و کنترل کنید.
- مدیریت تنظیمات (CM): درخواست‌های تغییرات مربوط به assetهای استفاده شده می‌توانند مربوط به assetهای متناظر قابل استفاده باشند. مدیریت تنظیمات دامنه و برنامه را همگام کنید.

#### ۴-۶-۳- سطح سوم: Defined

در این سطح، فرایندها با سازمان هم‌تراز شده‌اند و مهندسی در یک شکل منظم در سازمان انجام می‌شود. برای مهندسی خط تولید نرم‌افزار، این به معنی داشتن کنترل روی variability و assetهای قابل استفاده دوباره، هم در مرحله تولید و هم در مرحله استفاده است.

• مهندسی دامنه

مهندسی دامنه در سطح CMMI 3 انجام می‌پذیرد. گسترش‌هایی برای محیط‌های فرایند زیر لازم است:

- ایجاد نیازمندی‌ها (RD): برای یک بخش کسب و کار کامل، نیازمندی‌ها را برای چند محصول ایجاد کنید. حوزه خط تولید را تعریف کنید. محصولاتی که ساخته خواهند شد را مشخص کنید. اشتراک و تفاوت‌های آن محصولات را هم تعریف کنید.
- راه‌حل فنی (TS): باید variability در جنبه‌های عملی و سناریوهای دامنه شامل شود. معماری



- پلت فرم و بسته‌های داده فنی را ایجاد کنید. چندین منبع و مقصد برای رابطها در نظر بگیرید.
- واریسی (VE): اطمینان حاصل کنید که مهندسی برنامه از محصولات ساخته شده دامنه استفاده درستی می‌کند.
- ارزیابی (VA): مهندسی برنامه یک ذی‌نفع برای فرایند ارزیابی دامنه است.
- تمرکز فرایند سازمانی (OPF): پلت فرم دامنه، رویه‌های استفاده از این پلت فرم، متدولوژی‌ها، مؤلفه‌های قابل استفاده دوباره و رهنمودها را در نظر بگیرید. چندین محصول در یک بخش کامل کسب و کار را در نظر بگیرید. از حوزه خط تولید استفاده کنید.
- تعریف فرایند سازمانی (OPD): پلت فرم دامنه، رویه‌های استفاده از این پلت فرم، متدولوژی‌ها، مؤلفه‌های قابل استفاده دوباره و رهنمودها را در نظر بگیرید. چندین محصول در یک بخش کامل کسب و کار را در نظر بگیرید. از حوزه خط تولید استفاده کنید.
- آموزش سازمانی (OT): برای محصولات، فرایندهای برنامه و گروه‌های پروژه برنامه، آموزش در نظر بگیرید.
- مهندسی برنامه
- مهندسی برنامه در سطح CMMI 3 انجام می‌پذیرد. گسترش‌هایی برای محیط‌های فرایند زیر لازم است:

- ایجاد نیازمندی‌ها (RD): یک مشتری تنها در نظر گرفته می‌شود. قابلیت‌ها و variabilityهای خط تولید تولید نرم‌افزار به این مشتری ارائه می‌شود. از نیازمندی‌های خط تولید استفاده کنید، variabilityها را به موارد مناسب مقید کنید و نیازمندی‌های خاص برنامه را ایجاد کنید.
- راه‌حل فنی (TS): از assetهای دامنه استفاده کنید، variabilityها را مقید کنید و assetهای خاص برنامه را ایجاد کنید. معماری پلت فرم را برای برنامه تخصصی‌سازی کنید و از بسته داده فنی دامنه استفاده کنید.
- ارزیابی (VA): هر دو محصولات دامنه و برنامه را ارزیابی کنید. پرسنل را باید آموزش دهید تا بدانند که از assetهای دامنه چه استفاده‌ای باید کنند.
- آموزش سازمانی (OT): در مورد پلت فرم، استفاده از asset، فرایندهای دامنه و گروه‌های پروژه دامنه، آموزش دهید.

• همکاری

- همکاری در سطح CMMI 3 انجام می‌شود. گسترش‌هایی برای محیط‌های فرایند زیر لازم است:
- ایجاد نیازمندی‌ها (RD): نیازمندی‌های برنامه که بالقوه می‌توانند نیازمندی‌های خط تولید باشند را شناسایی کنید.
- راه‌حل فنی (TS): ملاک‌های انتخاب را مشخص کنید و گزینش از دامنه و قرار دادن در برنامه را هماهنگ کنید. Assetهای دامنه و برنامه فعلی و برنامه‌ریزی شده را به هم مرتبط کنید. Assetهای برنامه بالقوه برای دامنه را مشخص کنید. تصمیمات برای خرید، ساخت یا استفاده دوباره را هماهنگ کنید.

- یکپارچه‌سازی محصول (PI): یک نقشه راه برای محصولات آینده و توسعه‌های آینده محصولات فعلی تهیه کنید. پروتکل انتقال اصلی محصولات و زمان‌بندی انتقال محصولات را مشخص کنید. اجتماع بین مهندسی دامنه و برنامه را مشخص کنید.
- واری (VE): محیط واری دامنه، رویه‌ها و معیارها را به صورت همزمان و تکرارشونده با محیط واری برنامه ایجاد کنید. ارتباط بین نتایج واری و کارهای تصحیح کننده را با مهندسی دامنه و برنامه برقرار کنید. سیاست طرح‌ریزی را بین مهندسی دامنه و برنامه به اشتراک بگذارید.
- ارزیابی (VA): محیط ارزیابی دامنه، رویه‌ها و معیارها را به صورت همزمان و تکرارشونده با محیط ارزیابی برنامه ایجاد کنید. سیاست طرح‌ریزی را بین مهندسی دامنه و برنامه به اشتراک بگذارید.
- تمرکز فرایند سازمانی (OPF): اهداف کارایی فرایند سازمانی را روی تمام محصولات خط تولید مشخص کنید. طرح‌های اجرایی را بین مهندسی دامنه و برنامه همگام کنید.
- تعریف فرایند سازمانی (OPD): مسئولیت‌هایی که چندین پروژه و محصول را پوشش می‌دهند، تخصیص دهید.
- مدیریت ریسک (RSKM): راهبرد مدیریت ریسک و طرح‌های کاهش ریسک باید هر دو مهندسی برنامه و دامنه را پوشانند.
- ارزیابی و تحلیل تصمیم (DAR): اطمینان حاصل کنید که ارزیابی راه‌حل‌های دیگر، جنبه‌های هر دو دامنه و برنامه را می‌پوشاند.

#### ۴-۶-۴ - سطح چهارم: Quantitatively managed

در این سطح، فرایندها مدیریت می‌شوند و مهندسی در سازمان اندازه‌گیری می‌شود. برای مهندسی خط تولید نرم‌افزار، این بدین معنی است که کنترل کمی روی *variability* و *asset* های قابل استفاده، هم در ایجاد و هم در استفاده انجام می‌شود.

- مهندسی دامنه
- مهندسی دامنه در سطح 4 CMMI انجام می‌شود، و فرایندهای خط تولید نرم‌افزار در سطح 3 CMMI انجام می‌شود. گسترش‌هایی برای محیط‌های فرایند زیر لازم است:
- مدیریت پروژه کمی (QPM): در آمار، زیرفرایندهای مهندسی برنامه مربوطه را در نظر بگیرید.
- مهندسی برنامه
- مهندسی برنامه در سطح 4 CMMI و فرایندهای خط تولید نرم‌افزار سطح 3 انجام می‌شود. گسترش‌هایی برای محیط‌های فرایند زیر لازم است:
- مدیریت پروژه کمی (QPM): در آمار، زیرفرایندهای مهندسی دامنه مربوطه را در نظر بگیرید.
- همکاری

○ همکاری در سطح 4 CMMI و فرایندهای خط تولید نرم افزار سطح 3 انجام می شود. گسترش هایی برای محیط های فرایند زیر لازم است:

○ مدیریت پروژه کمی (QPM): وابستگی ها و رفتار همگام سازی فعالیت ها بین مهندسی دامنه و برنامه را اندازه بگیرید. ارتباط تأثیرات بین مهندسی دامنه و برنامه را برقرار کنید. فعالیت های بهبود روی کارایی پروژه که به پروژه های دیگر وابسته است را مذاکره کنید. ذی نفعان پروژه های برنامه و دامنه را مشخص کنید.

#### ۴-۶-۵- سطح پنجم: Optimising

در این سطح، فرایندها به صورت پیوسته در مورد تأثیرشان روی سازمان بهینه سازی می شوند. برای مهندسی خط تولید نرم افزار، این به معنی بهبود مشارکتی مهندسی دامنه و برنامه است.

- مهندسی دامنه:  
مهندسی دامنه در سطح 5 CMMI و فرایندهای خط تولید نرم افزار در سطح 4 انجام می شوند.
- مهندسی برنامه:  
مهندسی برنامه در سطح 5 CMMI و فرایندهای خط تولید نرم افزار در سطح 4 انجام می شوند.
- همکاری:  
همکاری در سطح 5 CMMI و فرایندهای خط تولید نرم افزار در سطح 4 انجام می شوند.

#### ۴-۶-۶- خلاصه

سطح های ارزیابی برای بعد فرایند برپایه CMMI قرار گرفته اند: Initial، Managed، Defined، Quantitatively managed و Optimising. در سطح Initial، هیچ فرایند خط تولیدی وجود ندارد. مهندسی دامنه و همکاری تقریباً در حد صفر است. بعد فرایند FEF با جنبه های زیر سروکار دارد:

- مهندسی دامنه  
این مورد از حد صفر تا حد تسلط بر فرایند پیش می رود. مهندسی دامنه از مشخص کردن اشتراکات، variability و پلت فرم قابل استفاده دوباره شروع شده و به برنامه ریزی و تعریف سیاست ها برای همه فرایندهای مدیریت برنامه ختم می شود.
- مهندسی برنامه

در پایین‌ترین سطح بلوغ که تنها فرایند ایجاد می‌باشد، رشد کرده می‌رسد به جایی که فقط شامل فرایندهای با اهمیت کم می‌شود ولی در نهایت در سطح بلوغ بالا به استفاده دوباره کردن از نه تنها asset های فنی، بلکه همه نوع سیاست‌ها می‌رسد.

- همکاری

در ابتدا اصلاً وجود ندارد و بعداً بخش مهمی از فرایندهایی می‌شود که هماهنگ‌سازی بین فعالیت‌های مهندسی برنامه و دامنه را انجام می‌دهد، که شامل فعالیت‌هایی برای هم‌تراز کردن فعالیت‌های دیگر و ارتباط برقرار کردن asset های موجود بین پروژه‌های مختلف می‌شود.

#### ۴-۷- بُعد سازمان

بعد سازمان با نگاشت واقعی نقش‌ها و مسئولیت‌ها به ساختار سازمانی سروکار دارد. در مهندسی خط تولید نرم‌افزار، بعد سازمان میزان مؤثر بودن توزیع مهندسی دامنه و برنامه روی سازمان را اندازه‌گیری می‌کند. سازمان دارای ساختارها و مسئولیت‌ها برای مهندسی دامنه و یا برنامه و همچنین نقش‌های پشتیبان و هماهنگ‌کننده است. به طور دقیق‌تر، سازمان مسئولیت‌ها را بین پلت‌فرم (مهندسی دامنه)، برنامه‌های متناظر (مهندسی برنامه) و هماهنگ‌سازی درونی، توزیع می‌کند.

جنبه‌های زیر هرکدام در بعد سازمان در مهندسی خط تولید نرم‌افزار، نقش ایفا می‌کنند.

- نقش‌ها و مسئولیت‌ها

سازمان تا چه حدی مسئولیت‌های مجزا و ارتباطات موجود در مهندسی خط تولید نرم‌افزار را خوب مدیریت می‌کند: نقش‌های مهندسی نامتمایز یا خاص خط تولید نرم‌افزار.

- ساختار

این بخش با ساختار سازمانی که نقش‌ها و مسئولیت‌ها را وارد عمل می‌کند، سروکار دارد، و شامل هر دو ساختار اصلی (چیزی که در چارت سازمانی دیده می‌شود) و ساختار ثانویه (معمولاً مجازی که در چارت سازمانی دیده نمی‌شود) می‌شود.

- تمهیدات همکاری

این بخش با حوزه ارزش‌های مشترک سروکار دارد و شامل همکاری در ساختارهای سازمانی اصلی و ثانویه است.

#### سطح‌ها

سطح‌های زیر برای بعد سازمان در مهندسی خط تولید نرم‌افزار، مشخص شده‌اند.

- ۱. Project
- ۲. Re-use
- ۳. Weakly connected
- ۴. Synchronised
- ۵. Domain engineering

در بخش‌های بعدی، هرکدام از این سطوح با جزئیات بیشتری شرح داده شده‌اند.

#### ۴-۷-۱ - سطح اول: Project

این سطح، سطح پایه است. سازمان، آماده انجام پروژه به صورت مهندسی تک‌سیستم، می‌شود. با توجه به مسائل سازمانی، وضعیت‌های معمول زیر را می‌بینیم:

- نقش‌ها و مسئولیت‌ها

فقط نقش‌های مهندسی برنامه، که همان نقش‌های مهندسی نرم‌افزار سنتی هستند، تعریف شده‌اند.

- ساختار

ساختار سازمان برپایه ایجاد تک‌سیستم‌ها ایجاد شده است.

- تمهیدات همکاری

منابع انسانی ممکن است بین پروژه به اشتراک گذاشته شوند ولی asset‌های نرم‌افزاری معمولاً به اشتراک گذاشته نمی‌شوند.

#### ۴-۷-۲ - سطح دوم: Re-use

در این سطح، پروژه‌ها از استفاده دوباره به عنوان یک فرصت استفاده می‌کنند. ابتدا بعضی asset‌های مشترک مشخص می‌شوند و سپس به شکل مؤلفه‌های قابل استفاده دوباره refactor شده یا توسعه داده می‌شوند.

- نقش‌ها و مسئولیت‌ها

هیچ نقش مهندسی دامنه دقیقاً مشخص شده‌ای وجود ندارد. متخصصان مهندسی برنامه روی مرزهای پروژه‌ها همکاری می‌کنند تا asset‌های مشترک را شناسایی کرده و به اشتراک بگذارند.

- ساختار

ساختار تمرکز بر انجام دادن پروژه‌ها دارد. بعضی منابع انسانی ارشد بر پیدا کردن و ایجاد مؤلفه‌های قابل استفاده دوباره تخصیص داده شده‌اند.

- تمهیدات همکاری  
همکاری برپایه مذاکرات و اشتراک اطلاعات بین پروژه‌ها هستند.

#### ۴-۷-۳ - سطح سوم: Weakly connected

در این سطح، یک یا چند سازمان جدای مهندسی دامنه و چند سازمان مهندسی برنامه وجود دارد. تعاملات ساده‌ای بین آن‌ها در فازهای ابتدایی و انتهایی چرخه حیات مهندسی برنامه و دامنه وجود دارد.

- نقش‌ها و مسئولیت‌ها  
نقش‌ها و مسئولیت‌ها برای هر دو بخش مهندسی دامنه و برنامه تعریف می‌شود. مسئولیت‌های تعریف شده‌ای بین سازمان مهندسی دامنه (برای assetهای پلت‌فرم) و سازمان‌های مهندسی برنامه (برای assetهای برنامه) وجود دارد.

- ساختار  
نقش‌های برنامه و دامنه در سازمان توزیع می‌شود. پروژه‌های مهندسی دامنه جدایی وجود دارند. هر دو مهندسی دامنه و برنامه اکثراً دارای ساختار دارای گرایش به پروژه هستند.

- تمهیدات همکاری  
همکاری برپایه اسناد است و اکثراً به شکل تعویض نیازمندی‌ها و مدیریت اشتراکی درخواست‌ها و خطاهای گزارش شده بین پروژه‌های مهندسی دامنه و چند پروژه مهندسی برنامه است.

#### ۴-۷-۴ - سطح چهارم: Synchronised

در این سطح، تعاملات متعددی بین مهندسی دامنه و مهندسی برنامه وجود دارد.

- نقش‌ها و مسئولیت‌ها  
نقش‌های هماهنگ کننده بین مهندسی برنامه و دامنه و بین سازمان‌های مهندسی دامنه وجود دارند. مهندسی دامنه نقش بزرگی در ایجاد نرم‌افزار دارد.

- ساختار  
یک ساختار ثانویه (مجازی) وجود دارد که شامل وظایف بین تیمی هست. معماری پایه، مهم‌ترین جنبه‌های ایجاد در سازمان را مشخص می‌کند.

- تمهیدات همکاری  
همکاری و همگامی قوی بین پروژه‌های مدیریت برنامه و دامنه، تیم‌های بینابینی، گروه‌های نیروی وظایف و غیره وجود دارد. ملاقات‌های (مجازی) منظمی بین افرادی که نقش‌های هماهنگی دارند، وجود دارد.

#### ۴-۷-۵- سطح پنجم: Domain engineering

در این سطح، جداسازی مهندسی دامنه و مهندسی برنامه به عنوان ساختار ثانویه و دامنه‌های عمل‌کردی، ساختار ثانویه را مشخص می‌کند.

- نقش‌ها و مسئولیت‌ها

مهندسی دامنه و برنامه مجتمع می‌شوند. مهمترین نقش‌ها، نقش‌های مهندسی دامنه هستند. خیلی از افراد در هر دو نقش‌های مهندسی برنامه و دامنه درگیر هستند.

- ساختار

ساختار برپایه نظم مهندسی دامنه، شکل داده شده است. آن‌ها به طور ثابت توسط مهندسی دامنه و برنامه مشخص نشده‌اند.

- تمهیدات همکاری

افراد می‌توانند در صورت نیاز نقش‌های مهندسی دامنه و برنامه را برعهده بگیرند.

#### ۴-۷-۶- خلاصه

سطح‌های ارزیابی برای بعد سازمان عبارتند از: Synchronised، Weakly connected، Re-use، Project، Domain engineering. در سطح اولیه، هیچ ساختار سازمانی برای انجام مهندسی خط تولید نرم‌افزار وجود ندارد. اگر انجام شود هم فقط برای یک قسمت خاصی است و قابل مشاهده برای کل سازمان نیست. بعد سازمانی FEF با جنبه‌های زیر سروکار دارد.

- نقش‌ها و مسئولیت‌ها

سازمان از یک مرحله اولیه که در آن هیچ نقش مهندسی دامنه‌ای وجود ندارد، شروع کرده و در ادامه نقش‌های مهندسی دامنه بیشتری می‌گیرد و به جایی می‌رسد که این نقش‌ها، تعیین‌کننده‌ترین نقش‌های سازمان می‌شوند.

- ساختار

سازمان از یک مرحله اولیه که ساختار توسط یک حالت بر مبنای پروژه مشخص می‌شود، شروع کرده و در ادامه، مهندسی دامنه، قسمت بیشتری از ساختار را تعیین می‌کند، البته ابتدا در ساختار ثانویه و سپس رفته‌رفته در ساختار اولیه.

- تمهیدات همکاری

سازمان از یک مرحله اولیه که هیچ همکاری سازمان یافته‌ای وجود ندارد، شروع کرده و در نهایت به سمت یک سازمان تعاونی می‌رود.

#### ۴-۸- اجرای ارزیابی

نتیجه‌ی اعمال ارزیابی FEF پروفایلی است که در آن برای هر یک از ابعاد BAPO، سطحی بین ۱ تا ۵ مشخص شده است. بهترین پروفایل برای یک واحد سازمانی، بسته به شرایط و موقعیت آن قابل تعریف است. به طور کلی، رسیدن به بالاترین سطح در هر چهار بعد، الزاماً بهینه نیست؛ زیرا در مواردی به دلایلی سطوح پایین‌تر را باید ترجیح داد. مثلاً در مواردی که دامنه به بلوغ کافی نرسیده، یا زیست‌چرخ<sup>۲۹</sup> تولید سامانه کوتاه و کوچک است، تلاش برای رسیدن به سطوح بالا اشتباه خواهد بود؛ به ویژه در بُعدهای B و O، چرا که در این شرایط انعطاف‌پذیری اهمیت بیشتری دارد. برای نمونه در بُعد B، در صورتی که سازمان در کسب‌وکار خود بیش‌تر از دیگران پی‌روی می‌کند، سطوح پایین‌تر مناسب‌تر است؛ همان‌طور که در صورتی که در کسب‌وکار خود کسوت رهبری داشته باشد، سطوح بالاتر مناسب‌تر خواهد بود.

با همه‌ی این اوصاف، چرا باید از FEF استفاده کنیم؟ سه دلیل زیر قابل تصور است:

- برای برآورد<sup>۳۰</sup> وضعیت سازمان؛ تا بدانیم سازمان چه قدر خوب کار می‌کند.
- برای مقایسه‌ی سازمان‌ها (به عنوان یک محک<sup>۳۱</sup>)
- برای کمک به برنامه‌های ارتقای سازمان

در مورد هر یک از این سه دلیل، در بخش‌های بعدی مفصلاً بحث خواهیم کرد.

#### ۴-۸-۱- FEF برای برآورد وضعیت

در این مورد سازمان یا برآوردکنندگان نباید از قبل بدانند که چه امتیازی به دست خواهد آمد؛ هر چند ممکن است شاخصی در مورد نتیجه وجود داشته باشد. این شاخص می‌تواند به عنوان یک هدف مورد استفاده قرار گیرد تا برآورد با توجه به آن انجام شود؛ به این ترتیب که در هر یک از چهار بُعد، ویژگی‌ها و جنبه‌های مختلف هر سطح (از سطح ۲ تا سطح شاخص) مورد بررسی قرار می‌گیرد. برای این منظور باید پرسش‌نامه‌ای تهیه شود که با کمک آن بتوان ارزیابی را انجام داد. زمانی می‌گوییم به سطحی در یک بعد رسیده‌ایم که تمام اجزای پرسش‌نامه‌ی مربوط به آن سطح رعایت شده باشند.

<sup>29</sup> lifecycle

<sup>30</sup> assessment

<sup>31</sup> benchmark



باید توجه داشت که FEF تنها برای ارزیابی تولید خانواده‌ی محصولات نرم‌افزاری (خط تولید) می‌تواند به کار رود و برای تک سامانه‌ها قابل استفاده نیست. برای ارزیابی و برآورد وضعیت تولید آن‌ها باید به سراغ روش‌های دیگر رفت. در بُعد P، برآورد دیگری مبتنی بر CMMI می‌تواند میزان بلوغ فرآیند عادی تولید نرم‌افزار را اندازه‌گیری کند. در ابعاد دیگر، روش ارزیابی معتبری وجود ندارد، و این به آن معناست که محتوای این ابعاد می‌تواند ارتقا یابد، حتی اگر به بالاترین سطح رسیده باشیم. برای نمونه‌ای از آن چه توسط FEF پوشش داده نمی‌شود می‌توان به موارد زیر اشاره کرد:

- در بُعد B: وجود یک برنامه‌ی کسب‌وکار خوب
- در بُعد A: وجود یک معماری خوب. این مورد را می‌توان از طریق یک برآوردکننده‌ی معماری معمولی، نظیر ATAM ارزیابی کرد.
- در بُعد O: وجود نظامی برای آموزش و ارزیابی کارکنان

بعد از اتمام برآورد وضعیت، تصمیم با خود سازمان است: ممکن است تصمیم بگیرد که خود را در یک بعد یا همه‌ی ابعاد ارتقا دهد، یا که از نتیجه‌ی ارزیابی راضی باشد و تصمیم بگیرد که کار خود به همین شیوه‌ی فعلی ادامه دهد.

#### ۴-۸-۲- FEF به عنوان محک

در صورتی که سازمان الف بخواهد خودش را با سازمان‌های ب، پ، و ... مقایسه کند، از FEF به عنوان محک استفاده خواهد کرد؛ البته مشروط بر آن که نتیجه‌ی ارزیابی سازمان‌های ب، پ، و ... فراهم باشد، مثلاً به شکلی منتشر شده باشد. حالت دیگر هم این است که سازمان‌های الف، ب، پ، و ... همه عضو یک شرکت بزرگ‌تر باشند. در این صورت ممکن است اطلاعات تنها در داخل شرکت قابل دسترسی باشد.

علاوه بر این، نتیجه‌ی ارزیابی سازمان ب (به طور مثال) می‌تواند به عنوان هدف و شاخص تعیین شود (مانند بخش قبل). به این ترتیب می‌توانیم بفهمیم که سازمان الف چه تفاوت‌هایی با سازمان ب دارد، به این معنی که در هر بُعد سازمان الف بهتر یا برابر یا بدتر از سازمان ب ظاهر شده است. در ابعادی که سازمان الف بهتر عمل کرده، برآورد می‌تواند در سطح عمل‌کرد سازمان ب متوقف شود، چرا که منظور در این‌جا تنها محک‌زدن و مقایسه‌کردن است.

#### ۴-۸-۳- FEF برای ارتقا

در این حالت، سازمان هدفی را تعیین می‌کند که در آینده می‌خواهد به آن برسد. آن‌گاه با انجام برآورد با توجه به هدف یادشده، نتیجه می‌گیرد که در چه مواردی نیاز به ارتقا وجود دارد. علاوه بر این ممکن است در مواردی که سازمان از هدف تعیین‌شده جلوتر است، نوعی بازخورد به بخش‌های مربوط ارایه شود تا از تلاش در آن موارد قدری بکاهند، و در عوض آن را در راه بهبود و ارتقای ابعاد دیگر صرف کنند.

ارتقا ممکن است به صورت تکاملی انجام شود، به این معنی که ابتدا هدف رسیدن به نتایجی (نتایج ارزیابی) دست‌یافتنی‌تر و نزدیک‌تر باشد، و پس از نیل به آن، رسیدن به نتایجی بلندپروازانه‌تر مد نظر قرار گیرد. مثلاً هدف ابتدا این باشد که در تمام ابعاد به سطح ۳ برسیم، و سپس هدف، رسیدن به سطح ۴ در ابعاد O و A در نظر گرفته شود.

#### ۴-۸-۴- ارتباط با دیگر رویکردها

ارزیابی مبتنی بر FEF تا حد زیادی مشابه ارزیابی مبتنی بر CMMI است: در هر سطح، ارزیابی با توجه به پرسش‌نامه‌ای رو به رشد، به نتایجی منتهی می‌شود که حاصل ارزیابی وضعیت است. با این حال این دو روش ارزیابی هم‌پوشانی نخواهند داشت؛ در عوض CMMI برای ارزیابی تک سامانه‌ها به کار می‌رود. بُعد P چارچوب FEF تنها به اندازه‌گیری توانایی لازم برای تولید و توسعه‌ی خانواده‌ی محصولات نرم‌افزاری می‌پردازد. با وجود این، FEF افزون بر CMMI، به ابعاد کسب‌وکار، معماری، و سازمان هم می‌پردازد.

چارچوب PLP یا چارچوب فعالیت‌های خط تولید نرم‌افزار، فهرستی از فعالیت‌های ضروری برای توسعه و تولید خانواده‌ی محصولات نرم‌افزاری است. البته PLP چارچوب ساخت‌یافته‌ای ارایه نمی‌کند که به کمک آن بتوان کار برآورد، محک، و ارتقای مداوم سازمان‌ها را انجام داد. با این حال هر کدام از فعالیت‌های چارچوب یادشده، در بخش مشخصی از FEF نقشی بسیار حیاتی ایفا می‌کنند؛ مثلاً:

- حوزه‌ی فعالیت‌های مرتبط با مهندسی نرم‌افزار در بعد معماری،

- حوزه‌ی فعالیت‌های مدیریت فنی در بعد فرآیند،

- و حوزه‌ی فعالیت‌های مدیریت سازمانی در بعد کسب‌وکار یا سازمان.

سنجشگر فنی خط تولید (PLTP) و مدل‌های اقتصادی، به ویژه برای سازمان‌هایی که در سطوح ۱ ابعاد چهارگانه هستند، قابل استفاده خواهند بود. این سازمان‌ها در حال تصمیم‌گیری برای رفتن یا نرفتن به سمت خط تولید هستند. چارچوب PLTP و مدل‌های اقتصادی به آن‌ها کمک می‌کنند تا از نقطه‌ی شروع، و جهت‌گیری برای رسیدن به سطوح ۲ در ابعاد چهارگانه، دید روشن‌تری داشته باشند.

#### ۴-۹- سازمان‌های پیچیده

واقعیت این است که بسیاری از سازمان‌هایی که در حوزه‌ی تولید خانوادگی محصولات نرم‌افزاری فعالیت می‌کنند پیچیده‌اند. توسعه و تولید بین واحدهای مختلفی توزیع شده، که این واحدها ممکن است در فضاهای مکانی متفاوت، یا حتی در نواحی زمانی گوناگونی پراکنده شده باشند. در این میان شرکت‌ها یا سازمان‌های دیگری نیز ممکن در کارها درگیر شوند؛ مثلاً از طریق برون‌سپاری، یا سایر قراردادهای بین شرکت‌ها. این وضعیت ارزیابی خانوادگی محصولات نرم‌افزاری را دشوار می‌کند، هرچند FEF «اصولاً» قابل اعمال است.

با این حال در عمل، FEF در مورد سازمان‌های پیچیده قابل اعمال نیست. علت این موضوع آن است که اتمام ارزیابی به درازا می‌کشد و از سوی دیگر واحدهای مختلف ممکن است بسیار متفاوت عمل کنند. به این ترتیب نتیجه به دشواری ممکن است از سطح ۱ در هر چهار بعد فراتر رود.

بنابراین چاره‌ای که وجود دارد آن است که ارزیابی روی واحدهایی با ابعاد معقول انجام شود. این واحدها می‌توانند اداره‌ها، بخش‌ها، شرکت‌های طرف قرارداد، یا حتی یک بخش مجازی از مجموعه‌ای از سازمان‌ها باشند. البته محدود کردن ارزیابی به این واحدها به این معنی است که تنها بخشی از مهندسی تولید خانوادگی محصولات می‌تواند ارزیابی شود، زیرا هر واحد تنها روی بخش مشخصی از کل کار متمرکز است. به طور معمول جنبه‌های مختلف کار به یک واحد واگذار نمی‌شود؛ مثلاً یک واحد عمدتاً به دامنه، یا عمدتاً به برنامه‌ی کاربردی اختصاص پیدا می‌کند.

اعمال FEF روی واحدهای سازمان‌های پیچیده، منتج به مشاهدات زیر در هر یک از ابعاد چهارگانه‌ی BAPO می‌شود:

- **B:** در بررسی این بُعد، ارتباط بین کسب‌وکارها در داخل سازمان مهم خواهد بود؛ به ویژه ارتباط واحد مورد بررسی، با دیگر واحدهای دیگر سازمان. عوامل خارج سازمان نیز در مواقع مقتضی مورد توجه قرار می‌گیرند.
- **A:** آن بخش از معماری مورد توجه قرار خواهد گرفت که واحد یادشده مسوول آن است، خواه با ایجاد یا استفاده از آن.
- **P:** بخشی از فرآیند که در واحد مذکور اجرا می‌شود مورد توجه قرار می‌گیرد.

• O: در این بُعد، سازمان‌های زیرمجموعه‌ی واحد یادشده، و نقش آن‌ها در ارتباط با دیگر بخش‌های سازمان مورد توجه خواهند بود.

#### ۴-۱۰- معماری ساخت‌یافته

در سازمان‌های پیچیده در بسیاری از موارد، خانواده‌ی محصولات نرم‌افزاری ساخت‌یافته‌اند (به صورت سلسله‌مراتبی)، یا این که خانواده بخشی از یک گروه<sup>۳۲</sup> است. در صورتی سلسله‌مراتب خواهیم داشت که تعداد زیادی از خانواده‌های محصولات از زیرساخت یکسانی استفاده کنند که به صورت خانواده‌ای از محصولات مدل شده است. این خانواده‌ها از اجزای یک‌دیگر استفاده می‌کنند، و همه‌ی آن‌ها نیز الزاماً این پلتفرم را مورد استفاده قرار نمی‌دهند. در عوض در یک گروه، چندین خانواده از بخش‌های مختلف یک پلتفرم بسیار گسترده مجدداً استفاده می‌کنند؛ بنابراین استفاده‌ی مجدد درون پلتفرمی است. با این حساب می‌توان گفت تقریباً سامانه‌ای وجود ندارد، یا به ندرت ممکن است وجود داشته باشد که از پلتفرم به طور کامل استفاده کند.

در حالت سلسله‌مراتبی، مهندسی برنامه‌ی کاربردی برای پلتفرم کلی، در واقع مهندسی دامنه برای یک خانواده‌ی محصولات است. در حالت گروهی، ممکن است گاهی یک واحد، هم مسوول بخش مشخصی از خانواده‌ی محصولات باشد، و هم بخشی از دامنه. در این حالت اعمال FEF کمی دشوارتر خواهد بود، هرچند ارزیابی به طور کلی به همان شیوه صورت می‌گیرد. هر واحد هم برای خانواده‌ای که مسوول آن است ارزیابی می‌شود، و هم در ارتباط با دیگران در کل ساختار. این ارزیابی باید به طور جداگانه برای هر نقشی که هر واحد در روند تولید و توسعه دارد انجام شود. این نقش‌ها بخشی از مهندسی دامنه یا برنامه‌ی کاربردی را دربرمی‌گیرند. به این ترتیب یک سازمان یا واحد سازمانی ممکن است به ازای هر یک از نقش‌هایی که ایفا می‌کند، به طرز ویژه‌ای ارزیابی شود، و به تبع آن نتیجه‌ی متفاوتی از ارزیابی به دست آورد. به این ترتیب می‌توان نتیجه گرفت که حتی در سازمان‌های غیرپیچیده نیز تنها بخشی از مهندسی خانواده‌ی محصولات بر حسب نقش، ارزیابی می‌شود. به این معنی که بسیاری از جنبه‌ها با توجه به نقش، در ارزیابی لحاظ نخواهند شد.

<sup>32</sup> population

## ۵- پرسشنامه ارزیابی خط تولید بر اساس مدل FEF

### ۵-۱- مقدمه

در فصل قبل، مدل FEF و تمامی سطوح ارزیابی آن به صورت جامع معرفی شد. هدف از این مدل تعیین میزان بلوغ خط تولید سازمان در هر یک از چهار حوزه کسب و کار، معماری، فرآیند و سازمان است. با این وجود این مدل صرفاً ساختاری توصیفی برای ارزیابی میزان بلوغ ارائه کرده است. برای آن که بتوان در عمل از این مدل استفاده کرد، نیاز به طراحی یک پرسشنامه با سؤالاتی دقیق است تا با دریافت پاسخ سؤالات آن از شرکت مربوطه، بتوان میزان بلوغ خط تولید آن شرکت را ارزیابی کرد.

بدین ترتیب بر پایه مدل FEF، پرسشنامه‌ای توسط نگارندگان این مقاله طراحی شد که در این فصل ارائه می‌شود. نتایج حاصل از این پرسشنامه در فصل بعد ذکر خواهد شد.

### ۵-۲- ربع کسب و کار (Business)

۱. استفاده از خط تولید نرم‌افزاری (SPL) در سیاست‌های تجاری شرکت شما به خصوص در بخش

بازاریابی، فروش و مدیریت محصولات چه جایگاهی دارد؟

a. آیا از مزایای آن آگاه هستید؟

b. آیا راهبرد مشخصی برای استفاده از آن دارید؟

c. آیا نرخ بازگشت سرمایه برای سرمایه‌گذاری در این بخش را محاسبه کرده‌اید و آن قدر

مناسب بوده که به عنوان پیش‌ران کسب و کار، مورد استفاده قرار گرفته باشد؟

d. آیا استفاده از SPL منجر به بهبود راهبردهای کسب و کار شما شده است؟

۲. میزان بودجه و سرمایه‌گذاری شما در SPL چه جایگاهی دارد؟

a. آیا برای تهیه یک مخزن از دارایی‌های قابل استفاده مجدد تلاش کرده‌اید؟

b. آیا بودجه مدون و معینی برای SPL در اختیار دارید؟

c. آیا به صورت کمی، بهره‌ای که از SPL بدست می‌آورد را محاسبه کرده‌اید؟

d. آیا بین هزینه‌های جاری و منافع بدست آمده از SPL در پیش‌بینی‌های آتی شما یکپارچگی

وجود دارد؟

۳. از دید مدیران سطح بالا، میزان تاثیر SPL چگونه دیده می‌شود؟

a. آیا مدیران سطح بالا نسبت به استفاده از SPL آگاهی دارند؟

b. آیا مدیران سطح بالا از SPL پشتیبانی جدی می‌کنند و استفاده از آن در چشم‌انداز کیفی

سازمان دیده می‌شود؟

- c. آیا استفاده از SPL به صورت کمی هم در چشم‌انداز سازمان دیده شده است؟
- d. آیا توسعه SPL روی چشم‌انداز هم تاثیر گذاشته است؟
۴. در برنامه های بلندمدت سازمان و نقشه راه آتی آن چه استفاده‌ای از SPL شده است؟
- a. آیا ارتباط خاصی بین سیستم‌های توسعه داده شده وجود دارد؟
- b. آیا برنامه‌های جداگانه‌ای برای مهندسی دامنه و برنامه وجود دارد؟
- c. آیا این برنامه‌ها با یکدیگر هماهنگ شده‌اند؟
- d. آیا از این برنامه‌ها و نقشه‌های راه برای ایجاد بهترین ارزش تجاری، از بعد استراتژیک استفاده‌ای می‌شود؟

### ۵-۳- ربح معماری (Architecture)

#### ۵-۳-۱- سطح ۱

- ۱-۱) آیا استفاده‌ی مجدد به شکل نظام‌مند صورت می‌گیرد؟
- ۲-۱) آیا معماری مشترکی بین محصولات تعریف شده است؟
- ۳-۱) آیا برای پشتیبانی از تغییرات در محصولات مشابه چاره‌ای اندیشیده شده است؟

#### ۵-۳-۲- سطح ۲

- ۱-۲) آیا استفاده‌ی مجدد مبتنی بر محصولات آماده‌ای است که از شرکت‌های دیگر تهیه می‌شود؟
- ۲-۲) آیا معماری محصولات مشابه مبتنی بر زیرساختی است که شرکت‌های دیگران فراهم کرده‌اند؟
- ۳-۲) آیا تغییرات قابل پشتیبانی در محصولات مشابه تنها محدود به بخش‌هایی است که زیرساخت یادشده پشتیبانی می‌کند؟

#### ۵-۳-۳- سطح ۳

- ۱-۳) آیا مجموعه‌ای از دارایی‌های (مولفه‌های) نرم‌افزاری را در قالب یک پلتفرم مجدداً استفاده می‌کنید؟
- ۲-۳) آیا معماری مرجعی برای تولید برنامه‌های کاربردی تعریف شده است و قواعدی نیز در کنار آن وجود دارد؟
- ۳-۳) آیا تغییرات قابل پشتیبانی را بر اساس معماری مرجع مشخص می‌کنید؟

#### ۵-۳-۴ - سطح ۴

۴-۱) آیا در مجموعه‌ی دارایی‌های نرم‌افزاری که مجدداً استفاده می‌شود، تغییرپذیری پشتیبانی شده است؟

۴-۲) آیا معماری مرجع ویژگی‌های معماری برنامه‌های کاربردی مشخص می‌کند و آیا راه حل‌هایی برای نیازمندی‌های کیفی (غیرعمل‌کردی) نیز در این معماری پیش‌بینی شده است؟

۴-۳) آیا معماری یادشده نقاط تغییر و گونه‌های قابل استفاده در معماری برنامه‌های کاربردی را مشخص خواهد کرد؟

#### ۵-۳-۵ - سطح ۵

۵-۱) آیا تغییرپذیری به شکل صریح در مجموعه‌ی دارایی‌های نرم‌افزاری لحاظ شده است و سازوکار پیکربندی نیز قابل تغییر است؟

۵-۲) آیا معماری مرجع، معماری برنامه‌های کاربردی را به طور کامل مشخص می‌کند و پیکربندی به شکل خودکار ممکن شده است؟

۵-۳) آیا تغییرپذیری به طور کامل با معماری مرجع یک‌پارچه شده، و با کمک مدل‌های یا زبان‌های استانداردشده در سازمان قابل توصیف است؟ آیا گونه‌ها به شکل خودکار قابل ساختن‌اند؟

#### ۵-۴-۴ - ربع فرآیند (Process)

#### ۵-۴-۱ - Domain Engineering

Initial:

۱. وجود Domain Activities در فرآیند مهندسی شرکت

Managed:

۲. مستندات پیرامون نیازمندی‌های خط تولید وجود دارد؟ (Requirement Engineering)

۳. آیا در این مستندات، به تنوع و Variability هم به نحوی پرداخته شده است؟

۴. آیا فعالیتهای مربوط به Domain در Plan نمود و تاثیر دارد؟

۵. آیا ارتباط بین Domain و Application در Plan نمود و تاثیر دارد؟

۶. آیا استفاده از مصنوعات دامنه مورد نظارت قرار می‌گیرد؟

۷. آیا دیدی کلی از وضعیت خط تولید موجود است؟  
۸. آیا Versioning بر روی مصنوعات خط تولید اعمال می شود؟

#### Defined

۹. آیا Scope خط تولید تعریف شده است؟  
۱۰. آیا مشخصاً چند محصول در یک شاخه بازار هدف گذاری شده است؟  
۱۱. آیا نیازمندیهای محصولات خط تولید، به نحوی تعریف شده است؟  
۱۲. آیا Commonality ها و Variability ها به صورت صریح تعریف شده اند؟  
۱۳. آیا محصولات خط تولید، به نحوی تعریف شده اند؟  
۱۴. آیا بستر معماری لازم برای دامنه در قالب یک راهکار فنی، تعریف شده و موجود است؟  
۱۵. آیا Variability ها در راهکار فنی ارائه شده برای بستر معماری، نمود دارند؟ (وجود Interface های مختلف تعریف شده و ...)  
۱۶. آیا Verification برای اطمینان از این که از مصنوعات دامنه در جای درست و به منظور صحیح استفاده شده است صورت می گیرد؟  
۱۷. آیا Validation بر روی مصنوعات دامنه برای اطمینان از این که معتبر و Valid هستند صورت می گیرد؟  
۱۸. آیا فرآیند سازمانی خط تولید تعریف شده و مستند است؟  
۱۹. آیا فرآیند آموزش برای محصولات نهایی یا استفاده از مصنوعات دامنه تعریف شده است؟

#### Quantitatively Managed

۲۰. آیا QPM وجود دارد؟  
۲۱. آیا معیارهای مدرج در قالب ابزار یا فرآیند منظم اندازه گیری یا ارزیابی می شوند؟  
۲۲. آیا بر روی نتایج آماری حاصل، تحلیل با روش مشخص (فرآیندی) صورت می گیرد؟

#### Optimizing

۲۳. آیا بهبود هدف گذاری شده، برای مساله ای که از روی معیارهای عمیقاً تعریف شده کشف یا مشخص گردیده است، وجود داشته است؟  
۲۴. آیا وجود دارد؟



۱. وجود فعالیتهای مهندسی محصول

Managed

۲. آیا نیازمندیها استخراج می شوند؟

۳. هم به صورت مستقل هم به صورت باز استفاده از نیازمندیهای دامنه؟

۴. آیا در Plan استفاده از مصنوعات دامنه دیده شده است یا تاثیری صریح در آن دارد؟

۵. آیا تحلیل ریسک وابستگی به فعالیتهای دامنه بر روی Plan انجام می شود؟

۶. آیا تاثیر مهندسی دامنه بر Scope پروژه ها وجود دارد؟

۷. آیا PMC وجود دارد؟

۸. آیا استفاده از مصنوعات دامنه در مهندسی محصول تحت Measurement یا تحلیل قرار دارد؟

۹. آیا بازاستفاده از محصولات دامنه تاثیر خود را در Versioning محصول می گذارد؟ یا مبنایی

برای Versioning محصول هست؟ یا تحت Versioning قرار می گیرند؟

Defined

۱۰. برای یک محصول یا مشتری خاص، نیازمندیها استخراج شوند. در حین استخراج نیازمندی

ها، از نیازمندی های موجود در دامنه استفاده مجدد شود. همچنین نیازمندی های محصول به

نیازمندی های دامنه مرتبط و متصل شوند. آیا چنین فرآیندی در مهندسی محصول تعریف

شده و وجود دارد؟

۱۱. آیا از محیط نرم افزاری تعریف شده ای برای استفاده مجدد از مصنوعات دامنه در فرایند

مهندسی محصول استفاده شده است؟

۱۲. Validation و Verification در فرآیند ایجاد به نحوی تعریف شده هم بر روی مصنوعات دامنه،

هم نسخه تنظیم شده آنها و هم مصنوعات ویژه محصول انجام می شود؟

۱۳. افراد فنی آموزش لازم برای استفاده مجدد از مصنوعات دامنه را دیده اند؟ یعنی آیا آگاه هستند

که از هر مصنوع دامنه چه کاربردی می توان گرفت؟

۱۴. آیا گروه های خاصی وجود دارد که آموزش دیده باشند که به صورت تیمی پیوند بین پروژه

خاص و دامنه را برقرار کنند و محقق سازند؟

Quantitatively Managed

۱۵. آیا QPM برای مهندسی محصول وجود دارد؟

۱۶. آیا معیارهای مدرج در قالب ابزار یا فرآیند منظم اندازه گیری یا ارزیابی می شوند؟

۱۷. آیا بر روی نتایج آماری حاصل، تحلیل با روش مشخص (فرآیندی) صورت می گیرد؟

### Optimizing

۱۸. آیا بهبود هدف گذاری شده، برای مساله ای که از روی معیارهای عمیقاً تعریف شده کشف یا

مشخص گردیده است، وجود داشته است؟

۱۹. آیا وجود دارد؟

### Collaboration – ۳-۴-۵

#### Initial

۱. آیا فعالیتی برای ارتباط و همکاری بین دامنه و محصول وجود دارد؟

#### Managed

۲. آیا ردگیری دوجهته بین نیازمندی های دامنه و محصول وجود دارد؟

۳. آیا طول عمر بیشتر مصنوعات دامنه به نسبت محصول در Plan دیده شده است؟

۴. آیا در Plan همگام کردن مهندسی دامنه و محصول دیده شده است؟

۵. آیا نظارتی بر ارتباط بین دامنه و محصول هست؟

۶. آیا نظارت و کنترلی برای همگامی مهندسی دامنه و محصول انجام می شود؟ (مثلا تعریف نقاط همگامی)

۷. آیا Versioning و Configuration Management در دامنه بر محصول یا بالعکس اثر دارد؟ آیا همگام هستند؟

#### Defined

۸. آیا فرآیند تعریف شده برای تکمیل احتمالی نیازمندی ها و دیگر مصنوعات دامنه از روی نیازمندی ها و مصنوعات محصولات وجود دارد؟

۹. آیا روش و معیار تعریف شده برای انتخاب از بین مصنوعات و برای مصنوعات چه در دامنه و چه در مهندسی محصول وجود دارد؟ آیا این امر با ابزار و محیط ایجاد یکپارچه شده است؟

۱۰. آیا تصمیم سفارش و خرید، استفاده مجدد یا ساخت از نو، به صورت صریح برای مصنوعات، تصمیم سازی و تصمیم گیری می شود؟
۱۱. وجود نصب العین و نقشه راه برای محصولاتی که در آینده قرار است به مجموعه خط تولید افزوده شود یا از طریق خط تولید، ایجاد گردند.
۱۲. تعریف زمان بندی و توالی که برای تحویل و انتقال محصول و فازها به مشتری نیاز است.
۱۳. امکان سازهمبندی مصنوعات دامنه و محصول
۱۴. داشتن توامان و سازگار و هماهنگ Verification و Validation در دامنه و محصول
۱۵. تبادل نتایج Verification و Validation بین دامنه و محصول
۱۶. به اشتراک گذاری یک سیاست Planning بین دامنه و محصول.
۱۷. اهداف و اهداف اجرایی و سطح عملکردی مورد انتظار سازمان از خط تولید تعریف شده باشد.
۱۸. همگام سازی برنامه های عملیاتی بین دامنه و محصول.
۱۹. مسئولیتهایی که بین چندین پروژه یا محصول مشترک باشد در سازمان تعریف شده باشد. در واقع در فرآیند سازمانی، ارتباط بین دامنه و محصول در قالب نقش های سازمانی تعریف شده باشد.
۲۰. برقراری ارتباط و مدیریت بین مصنوعات و محصولات فعلی و قبلی و آتی در دامنه و محصولات
۲۱. وجود فرآیند تعریف شده برای تشخیص مصنوعات جدید برای دامنه از روی محصولات
۲۲. مدیریت ریسک و راهبرد آن هم پیرامون دامنه و هم محصول موجود و مناسب باشد.
۲۳. DAR هم از منظر دامنه و هم محصول صورت بگیرد.

### Quantitatively Managed

۲۴. مدیریت QPM برای وابستگی های بین دامنه و محصول و نحوه پیشبرد و پیشرفت در فعالیتهای همگام سازی
۲۵. برقراری ارتباط برای بازخورد بین دامنه و محصول
۲۶. بررسی گلوگاه های مابین این دو
۲۷. راهبری و نظم بخشی به افراد در هر دو حوزه و با دید توامان

### Optimizing

۲۸. آیا بهبود هدف گذاری شده، برای مساله ای که از روی معیارهای عمیقاً تعریف شده کشف یا مشخص گردیده است، وجود داشته است؟
۲۹. آیا وجود دارد؟

## ۵-۵-۵- ریع سازمان

### ۵-۵-۱- سطح اول

۱. آیا تنها شاهد نقش‌هایی برای ساخت سیستم‌ها به شکل منفرد هستیم یا مدیریت و مهندسی دامنه (پلت‌فرم و بسترهای برنامه‌ها) وجود دارد؟
۲. آیا ساختار سازمان (چارت سازمانی) تنها برای ایجاد سیستم‌ها به شکل منفرد طراحی شده است؟
۳. آیا منابع انسانی بین پروژه‌های مختلف به اشتراک گذاشته می‌شوند؟ (یعنی مثلاً یک نفر در بیش از یک پروژه کار می‌کند)
۴. آیا دارایی‌های نرم‌افزاری (assetها) بین پروژه‌های مختلف به اشتراک گذاشته می‌شوند؟

### ۵-۵-۲- سطح دوم

۱. آیا نقش‌هایی به صورت واضح (با تعریف دقیق و مشخص و کامل) برای مدیریت و مهندسی دامنه تعریف شده است؟
۲. آیا منابع انسانی برای ایجاد مؤلفه‌های قابل استفاده دوباره (reusable components) اختصاص داده شده است؟
۳. آیا همکاری بین پروژه‌های به شکل مذاکرات و اشتراک مطالب انجام می‌شود؟

### ۵-۵-۳- سطح سوم

۱. آیا نقش‌های مربوط به دامنه و برنامه‌ها به شکل جدا و به صورت توزیع شده در تمام قسمت‌های چارت سازمانی وجود دارد؟
۲. آیا اشتراک اطلاعات بین پروژه‌های و همچنین بین برنامه‌ها و دامنه توسط مستندات انجام می‌شود؟

### ۵-۵-۴- سطح چهارم

۱. آیا نقش‌هایی تعریف شده است که کارشان هماهنگی بین فعالیت‌های مهندسی برنامه‌ها و مهندسی دامنه باشد؟
۲. آیا علاوه بر چارت رسمی سازمانی، ساختار فرعی و مجازی دیگری هم در سازمان وجود دارد که به صورت بین عملگری (cross functional) کار می‌کنند؟
۳. آیا به منظور اشتراک اطلاعات بین پروژه‌ها و همچنین بین دامنه و برنامه‌ها، میتینگ‌های منظمی برگزار می‌شود؟

۱. آیا مسئولیت‌های افراد در سازمان، مرتبط و وابسته به بخش‌های وظیفه‌مندی ( Functional Domains) در معماری کلی، تعریف شده‌اند؟
۲. آیا ساختار اصلی به طور غالب ناظر بر مهندسی دامنه است و نقش‌های مربوط به مهندسی برنامه در ساختار ثانویه و مجازی قرار گرفته است؟
۳. آیا افراد در صورت نیاز نقش‌های دامنه و برنامه می‌گیرند؟ (یعنی اگر لازم شد نقش عوض می‌کنند و نقش دیگری برعهده می‌گیرند، در مقابل این حالت که نقش‌ها برای افراد مشخص و ثابت و معمولاً غیرقابل تغییر باشد)

## ۶- نتایج ارزیابی خط تولید شرکت اعوان با استفاده از مدل FEF

### ۶-۱- مقدمه

در این فصل نتایج گفتگوها و جلساتی که بین نگارندگان این مقاله و مسئولان شرکت اعوان پیرامون مدل خط تولید، رویکردهای متفاوت به خط تولید، تاریخچه تشکیل خط تولید در شرکت اعوان، برنامه‌های کنونی اعوان برای گسترش خط تولید، مشتریانی که از خط تولید اعوان استفاده کرده‌اند و بسیاری مطالب جذاب و خواندنی دیگر که پیرامون خط تولید نرم‌افزاری مطرح شد ذکر می‌شود.

پس از توضیحی که مسئولان شرکت اعوان پیرامون خط تولید خود دادند، پرسش‌نامه‌ای که در فصل قبل آماده شده بود مورد سوال قرار گرفت و بر اساس پاسخ‌های دریافتی، میزان بلوغ شرکت اعوان در هر یک از ربع‌های مدل FEF ارزیابی شد که در ادامه نتایج این ارزیابی نیز ذکر خواهد شد.

منبع نوشته‌های مربوط به این فصل، عمدتاً گفتگوهای شفاهی است که در جلسات مختلف مورد بحث قرار گرفت و صدای آن ضبط شد. با این وجود احتمال خطا در انتقال مطالب وجود دارد که بابت خطاهای احتمالی پیشاپیش عذرخواهی می‌شود.

### ۶-۲- موارد کلی پیرامون خط تولید شرکت اعوان

- در شرکت اعوان در یکی/دو پروژه آغازین بخشی به نام core وجود نداشت. با گسترش فعالیت‌های شرکت و مشاهده وجود مشترکات فراوان میان پروژه‌ها، نیاز به وجود یک بخش مشترک احساس شد که برای پاسخ‌گویی به این نیاز، به صورت تکراری/افزایشی<sup>۳۳</sup> بخش core تشکیل شد. از دید مدیران این شرکت، تجربیات نخست نرم‌افزاری برای کسب آمادگی ورود به بازار مطرح شده بود ولی از ابتدا راهبرد شرکت اعوان بر داشتن یک خط تولید کارآمد تعریف شده بود. تحلیل، طراحی و پیاده‌سازی بخش core و تعریف نیازمندی‌های لازم برای ایجاد این دارایی مشترک، عاملی بود که به صراحت در سال‌های آغازین تاسیس شرکت انجام شد و بدین ترتیب شرکت اعوان یک شرکت مبتنی بر خط تولید محسوب می‌شود.
- خط تولید شرکت اعوان بر اساس نیازهای تجاری شکل گرفته است. در نتیجه ممکن است به صورت کامل با فضای نظری حاکم بر مقالات دانشگاهی مطابقت نداشته باشد. به عنوان مثال، اصطلاحی که بیشتر بیانگر خط تولید اعوان است، اصطلاح دسته خانواده محصولات نرم‌افزاری است. ضمناً مدل خط تولید این شرکت از تعریف صریح نقاط تغییر و متغیرها پشتیبانی نمی‌کند.

<sup>۳۳</sup> iterative/incremental

- عمده مولفه‌های که از آن به عنوان مولفه مرکزی و سنگ‌بنای خط تولید اعوان یاد می‌شود، مولفه WISE و مشخصاً بخش core از پروژه wise است.
- نه تنها از wise برای تولید و فروش محصولات مبتنی بر وب استفاده شده است، بلکه مستقیماً خود wise نیز به فروش رفته است.
- در چارچوب wise از نسخه‌بندی (versioning) استفاده شده است و نسخه‌های متعددی از آن تا به حال تولید شده است.
- در حال حاضر توسعه هر دو بخش core و برنامه‌ها به صورت موازی صورت می‌گیرد.
- عمده محصولات تولیدی اعوان به صورت نرم‌افزارهای مبتنی بر وب است. بخش تولید کننده کد، تنها محصولی است که خودش به صورت مبتنی بر وب نیست.
- در شرکت اعوان تجربیات قابل توجهی در زمینه فریمورک‌نویسی ایجاد شده است. مثلاً نوشتن فریمورکی برای ایجاد Smart TV ها
- در فریمورک نویسی بسیاری از مفاهیم مانند ایجاد یک متامدل با بحث خط تولید مشترک است.
- ابزارهای مورد استفاده در اعوان از روش معماری مبتنی بر مدل (Model Driven Architecture) استفاده می‌کنند. ضمناً از ابزار EMF که توسط eclipse توسعه داده شده است استفاده می‌شود. در MDA گفته می‌شود که صرفاً به زبان‌های موجود بسنده نشود، بلکه بر اساس نیازمندی‌ها به تعریف زبان‌های جدید پرداخته شود. مثلاً برای طراحی واسط کاربری، یک متامدل جدیدی تعریف می‌شود که در گام اول، قواعد حاکم بر این زبان مثل ساختارهای سازنده آن تعریف می‌شود. خدمتی که EMF ارائه می‌کند، فراهم کردن یک متامدل در قالب ecore است. در مرحله بعد می‌توان از ابزارهای خودکاری مانند model editor استفاده کرد تا به تعریف نقاط تغییر و متغیرهای قابل تعریف روی این خط تولید پرداخت. به عنوان مثال می‌توان از یک متامدل با سطح تجرید بالا مثل توصیفات معماری به یک مدل سطح پایین مثل مدل فیزیکی در قالب کد رسید. بنابراین کافی است که نقاط تغییر را در اثنای این تبدیلات به درستی تعریف کرد تا کد اجرایی متناسب با این نیازمندی تولید شود. در واقع این‌جاست که قدرت تولید نرم‌افزار با استفاده از مفاهیم خط تولید مشخص می‌شود.
- از ابزارهای موجود در پروژه متن باز Open Architecture Ware (OAW) برای این منظور استفاده می‌شود.
- البته در حال حاضر تمامی گام‌های تبدیل کد به صورت خودکار صورت نمی‌گیرد. بلکه صورت مساله به صورت دستی تعریف می‌شود که مصنوعات این گام با مصنوعات مدل UML تطابق دارد.

- روشگان مورد استفاده برای تولید محصولات در شرکت اعوان بدین نحو است: ابتدا نیازمندی‌ها در قالب سند Stakeholder Request جمع‌آوری می‌شود. مرحله بعد تهیه mission و vision است. سپس از روی این چشم‌انداز، تعدادی مورد کاربرد استخراج می‌شود. سپس بر روی این موارد کاربرد طبقه‌بندی صورت می‌گیرد که بر اساس شناختی است که از ابزارهای موجود ایجاد شده است. به عنوان مثال یک دسته از موارد کاربرد مربوط به گزارشات است که برای تحقق آن، ابزارهای خودکار وجود دارد. در فاز بعدی بحث استفاده از مفاهیم مطرح در خط تولید پیش می‌آید. بدین صورت که بسته‌هایی که امکان استفاده مجدد از آن‌ها وجود دارد، در قالب پروژه core دسته‌بندی می‌شوند.
- دو رویکرد کلی برای تولید یک خط تولید وجود دارد:

○ رویکرد بالا به پایین (top-down) : در این رویکرد که در مباحث کتاب رسمی درس به صورت گسترده به آن پرداخته شده است، ابتدا با مهندسی دامنه و انجام فعالیت‌های تحلیل، طراحی، پیاده‌سازی و تست بر روی دارایی‌های مشترک و کلیدی مطرح در یک دامنه خاص از صنعت، یک چارچوب<sup>۳۴</sup> آماده می‌شود که شامل تمامی بخش‌های مشترک میان تعدادی برنامه است. هر کدام از نقاط افتراق محصولات خروجی، در قالب نقاط تغییر<sup>۳۵</sup> شناسایی می‌شوند و مقادیری که قابل تخصیص به آن‌هاست در قالب متغیر<sup>۳۶</sup> تعریف می‌شود. به عنوان مثال در صنعت تلفن همراه، شرکت‌هایی که قصد ورود به بازار این صنعت را دارند، ابتدا روی ایجاد یک دارایی‌های پایه<sup>۳۷</sup> سرمایه‌گذاری می‌کنند که شامل ویژگی‌های مشترک میان تمامی انواع تلفن‌های همراه است مانند قابلیت برقراری مکالمه، نمایش شماره بر روی صفحه نمایشگر و ... در گام بعد برای هر مدل از تلفن همراه، نقاط تغییر مانند رنگ، اندازه، کیفیت دوربین و ... شناسایی می‌شوند. در نهایت برای هر کدام از این نقاط تغییر، مقادیر قابل قبول در قالب متغیر مشخص می‌شود که به عنوان مثال رنگ مشکی، رنگ صورتی و ... برای نقطه تغییر رنگ مطرح است.

○ رویکرد پایین به بالا (bottom-up) : در این رویکرد که در شرکت اعوان به عنوان رویکرد غالب مطرح است، از تعدادی دارایی عادی که در خلال توسعه نرم‌افزارها و برنامه‌های قبلی ایجاد شده است استفاده می‌شود و با تجمیع اشتراکات آن‌ها در یک پروژه مرکزی به نام core، یک فریمورک ایجاد می‌شود.

platform<sup>۳۴</sup>  
variation point<sup>۳۵</sup>  
variant<sup>۳۶</sup>  
core asset<sup>۳۷</sup>



- از دید شرکت اعوان در میان سازمان‌های کارفرما، خط تولید به عنوان یک نیاز مطرح شده است ولی لزوماً هنوز به عنوان راه‌حل شناخته نشده است. در نتیجه شرکت اعوان ایجاد یک ابزار تولید خط تولید را به عنوان مزیت رقابتی خود مطرح کرده است تا در هنگام بازاریابی محصولات بتواند بر این ویژگی ممتاز خود تکیه کند.
- شرکت اعوان داشتن خط تولید را به عنوان عاملی موثر در کاهش زمان توسعه برنامه‌ها، کاهش هزینه تولید محصولات و افزایش کیفیت برنامه‌ها و ... می‌داند. بدین ترتیب داشتن یک خط تولید کارآمد و به تبع آن کاهش هزینه‌های تولید محصولات جدید، منجر به افزایش احتمال موفقیت شرکت برای برنده شدن در پروژه‌های مناقصه‌ای محسوب می‌شود.
- شرکت اعوان خود را مصرف‌کننده خط تولید خود و نیز فروشنده خط تولید موجود می‌داند. بدین معنا که توانسته است خط تولید خود را نیز به صورت کامل برای استفاده سایر سازمان‌های متقاضی به فروش برساند. حتی دانش فنی استفاده از این خط تولید نیز در برخی موارد به سازمان‌های متقاضی واگذار شده است.
- یکی از ایرادات عمومی توسعه نرم‌افزار و به خصوص در حوزه خط تولید، فرسوده شدن خط تولید با گذر زمان و ارائه فناوری‌های جدید است که منجر به از رده خارج شدن فناوری‌های فعلی می‌شود. رفع این مشکل نیازمند تمهیدات خاص می‌باشد.
- راهبردهای این شرکت در حوزه خط تولید به صورت کلی مکتوب شده ولی به صورت شفاف اعلام عمومی نشده است. بنابراین قطعاً باید گامی برای اعلام عمومی راهبردهای استفاده از خط تولید صورت گیرد تا مشتریان و حتی کارمندان این شرکت جایگاه خود را در مدل خط تولید اعوان شناسایی کنند.
- در حال حاضر راهبرد شرکت اعوان برای قبول پروژه‌های جدید، میزان اشتراک‌پذیری آن با پروژه‌های قبلی و یا پروژه‌های آتی است. در نتیجه رویکرد اساسی و بسیار پر اهمیت در این شرکت، افزایش بیشینه قابلیت استفاده مجدد از محصولات و پروژه‌ها است که هم‌راستایی فراوانی با رویکرد و مباحث مطرح در خط تولید دارد.
- برخی از پروژه‌هایی که شرکت اعوان پذیرفته است، مستقیماً به توسعه core باز می‌گردد. این پروژه‌ها یک منفعت دو جانبه هم برای کارفرما و هم برای شرکت اعوان محسوب می‌شود.
- برنامه آتی شرکت اعوان در مورد خط تولید، ایجاد سامانه‌های سازمانی است. دو برنامه توسعه برای این منظور تعریف شده است. یکی از این برنامه‌ها که موضوع سال جاری است، ایجاد یک روش بالغ‌تر برای پاسخ‌گویی به نیازمندی‌های سازمانی است که استفاده از گذرگاه سرویس سازمانی (ESB) در این دامنه می‌گنجد. برنامه توسعه دیگر، ارائه سرویس‌های عمومی است. مثلاً ارائه یک

سرویس آموزشی یا یک شبکه اجتماعی که تمام کاربران اینترنتی امکان بهره‌مندی از این سرویس را دارند.

- ارتباط بین شبکه اجتماعی به عنوان یک سرویس عمومی با مفهوم خط تولید بدین صورت است که ابتدا ویژگی‌های مشترک میان شبکه‌های اجتماعی گوناگون نیز در قالب یک دارایی مرکزی، توسعه داده می‌شود و سپس برای هر یک از مصارف خاص، بخش مربوط به آن ایجاد می‌شود. بدین ترتیب مشابه رویکرد بالا به پایین، یک شبکه اجتماعی مبتنی بر خط تولید ایجاد می‌شود.
- نکته قابل توجه برای اطلاق رویکرد بالا به پایین به روش فوق در این است که پیش از ساخت محصول، باید نقاط تغییر آن مشخص شده باشد. در واقع باید گام‌های مهندسی دامنه و مهندسی برنامه به صورت مشخص طی شده باشد تا عنوان رویکرد بالا به پایین به آن اطلاق شود.
- در حال حاضر در شرکت اعوان تیم‌های جداگانه‌ای برای توسعه CORE و توسعه برنامه‌ها وجود دارد ولی این تیم‌ها به شدت به یکدیگر وابسته هستند و نیز با یکدیگر هماهنگ شده است.
- توسعه بخش CORE از دو حوزه نشات می‌گیرد. گاهی بر اساس نیازمندی‌های جدید فناوری مانند ارائه نسخه جدیدی از یک مولفه نرم‌افزاری مانند GWT، نیاز به انجام تغییرات مطابقت دهنده در CORE احساس می‌شود. گاهی نیز بر اساس نیازهای پروژه‌های مختلف که یا از جانب کارفرما مشخص شده و یا در خلال توسعه برنامه‌های جاری احساس می‌شود، بخش CORE دستخوش تغییرات خواهد شد.
- یکی از موارد مهم دیگر در حوزه مهندسی خط تولید، برقراری هماهنگی میان فعالیت‌های انجام گرفته برای توسعه CORE و فعالیت‌های لازم برای توسعه برنامه‌هاست. در واقع وجود یک نقشه راه از چگونگی برقراری این هماهنگی و سنگ‌نمایی از زمان‌های این هماهنگ‌سازی امری ضروری به نظر می‌رسد. البته با توجه به کوچک بودن شرکت اعوان از نظر تعداد نیروی کاری، شاید نیاز به بیان صریح این نقشه راه ضروری نباشد.
- بحث «شخصی‌سازی گسترده»<sup>۳۸</sup> در خط تولید شرکت اعوان منظور شده است. بدین معنا که برای هر کدام از مشتریان عمده محصولات خط تولید، شخصی‌سازی‌های مورد نیاز آن‌ها در نظر گرفته می‌شود. به عنوان مثال در پرتال جامع اعوان که به عنوان نقطه ورود به سامانه محسوب می‌شود، بر اساس نیاز هر مشتری، لوگوی مربوط به همان مشتری نمایش داده می‌شود. این مورد مثال مناسبی از تعریف لوگو به عنوان یک «نقطه تغییر» و مقداره‌ی آن لوگو به آرم شرکت متقاضی به عنوان «متغیر» می‌باشد.

- یک کارخانه ماشین‌سازی را در نظر بگیرید. برخی از فرآیندهای تولید تمامی ماشین‌ها با یکدیگر مشابهت دارند. مثلاً ماشین‌های سواری از مدل لاستیک یکسانی استفاده می‌کنند. بنابراین می‌توان فرض کرد که در خط تولید ایجاد تمامی ماشین‌های سواری، یک مرحله مشترک وجود دارد که در آن لاستیک به ماشین اضافه می‌شود. اما پس از خروج از این مرحله، هر کدام از ماشین‌ها وارد مرحله جداگانه‌ای می‌شوند. مثلاً هر کدام برای رنگ شدن بدنه به واحد جداگانه‌ای هدایت می‌شوند. در بحث خط تولید شرکت اعوان نیز این مثال صادق است. متدلوژی توسعه نرم‌افزار این شرکت به نحوی تعریف شده است که با تعریف نقاط تصمیم، فرآورده‌های تولید شده در هر گام به واحد مناسب در گام بعدی هدایت می‌شود.

- در بعد سازمان به بررسی ارتباط میان اعضای تیم تولید کننده core و تولید کننده برنامه‌ها پرداخته می‌شود. اگر این ارتباط به خوبی برقرار نشود، به اصطلاح چرخ چند بار اختراع می‌شود یعنی اعضای تیم تولید کننده برنامه بدون آن که از وجود یک مولفه در بخش core آگاه باشند، مجدداً شروع به تولید آن مولفه می‌کنند. در نتیجه شاهد مولفه‌هایی خواهیم بود که همگی برای پاسخ‌گویی به یک هدف واحد ولی هر کدام در یک ساختار مجزا تولید شده‌اند.

- **مستند سازی اطلاعات:** در این بخش به این مورد پرداخته می‌شود که با چه ساز و کاری افراد حاضر در تیم توسعه را مجبور می‌کنید که مولفه‌های تولید شده را مستند کرده و به اطلاع سایرین برسانند تا توسط سایر اعضای تیم مورد استفاده قرار گیرد. در شرکت اعوان، افرادی که بخش core را تولید می‌کنند، مستندات کاربری آن را نیز در قالب JavaDoc و یا مستنداتی که نشان‌دهنده چگونگی استفاده از این قابلیت است ارائه می‌کنند. مکانیزم بعدی، استفاده از تالارهای گفتگو در نشانی [asta.ir/forum](http://asta.ir/forum) می‌باشد. در این تالار گفتگو، مشکلات مطرح برای استفاده از مولفه‌های مختلف طرح شده و از جانب کاربران باتجربه‌تر، پاسخ داده می‌شود. در واقع از برنامه‌نویسان خواسته می‌شود تا پس از کشف یک راه‌حل آن را در تالار ثبت کنند. به علاوه، اسلایدهای نمایشی متعددی آماده شده است که به هنگام اضافه شدن عضو جدید به تیم، به وی ارائه می‌شود و در آن توضیح جامعی در مورد معماری خط تولید شرکت اعوان ذکر شده که در تهیه این گزارش نیز از این اسلاید نمایشی استفاده شده است.

- **برگزاری میتینگ بین اعضای core و برنامه:** برای برقراری هماهنگی میان ذینفعان مختلف یک پروژه نرم‌افزاری، مانند مشتریان و استفاده‌کنندگان از wise، شرکت اعوان به عنوان توسعه‌دهنده اصلی wise، نمایندگان شرکت‌های فراهم‌کننده زیرساخت و ... جلسات هفتگی برگزار می‌شود که در آن‌ها در مورد مطالب گوناگونی مانند نیازمندی‌های جدید، پرسش از امکانات موجود، چگونگی شخصی‌سازی برخی امکانات و ... سوال پرسیده می‌شود.

- ساختار سازمانی بین عملکردی و پذیرفتن نقش‌های متقابل: افرادی که به صورت دو به دو، نقش خود را در پروژه core با پروژه‌های سایر برنامه‌ها مبادله می‌کنند. مثلاً زمانی که وظایف یک فرد در برنامه پایان یابد، تعدادی وظیفه مبتنی بر core به وی تخصیص داده می‌شود. همین‌طور زمانی که پروژه‌های برنامه‌ای به پایان مهلت خود نزدیک می‌شوند، از اعضای تیم توسعه دهنده core برای تسریع در خاتمه آن پروژه استفاده می‌شود. بدین ترتیب تجارب میان اعضای این دو دسته تیم با یکدیگر به اشتراک گذاشته می‌شود.
- ارزیابی شرکت اعوان بر اساس مدل FEF صورت خواهد گرفت که در چهار ربع مطرح شده در مدل BAPO به تعیین میزان بلوغ شرکت در قالب یکی از سطوح یک تا پنج می‌پردازد. از این لحاظ مدل FEF با مدل CMMI مشابهت دارد و برای ارزیابی بعد فرآیند مستقیماً از مدل CMMI استفاده می‌کند ولی صرفاً روی این بعد تمرکز نمی‌کند. در ادامه سوالات پرسش‌نامه طراحی شده، مطرح و نتایج آن ارزیابی خواهد شد.

### ۶-۳- ارزیابی ربع کسب و کار

یکی از مباحث مطرح برای ارزیابی میزان بلوغ شرکت در استفاده از خط تولید، داشتن مدل‌های کمی است که با ارائه عدد و رقم‌های آماری از میزان سرمایه‌گذاری در بخش خط تولید و میزان منفعت‌های مالی بدست آمده از این سرمایه‌گذاری، به صورت روشن نسبت سود حاصله را محاسبه کند و در سایر تصمیم‌گیری‌های مدیریتی، از این معیار استفاده کند. مثلاً چه بخش‌هایی را وارد core کند و چه بخش‌هایی ارزش توسعه در حوزه مشترک را ندارد. متأسفانه شرکت اعوان تا کنون برنامه خاصی را برای ارزیابی کمی مدل خط تولید خود اجرا نکرده است. علت این اتفاق این بوده که تا به حال ارزیابی مالی میزان سرمایه‌گذاری در خط تولید، یک دغدغه جدی برای مدیران این شرکت نبوده است؛ زیرا ایشان باور داشتند که سرمایه‌گذاری در این حوزه، کاملاً دارای سود مالی و بازگشت سرمایه است. در واقع از آنجایی که شرکت اعوان شرکتی دانش‌بنیان و نه سرمایه‌محور محسوب می‌شود، در تصمیم‌گیری‌های خود کمتر به دنبال محاسبه بار مالی پروژه می‌باشد. با این وجود در چند سال اخیر، این شرکت نیز به محاسبه هزینه‌ها و درآمدهای هر پروژه پرداخته است و اجرای این کار را نیز بسیار مفید می‌داند و تا حدی این محاسبه منجر به تغییر در مدل کسب و کاری این شرکت شده است.

بدین ترتیب با توجه به استفاده شرکت اعوان از رویکرد پایین به بالا به عنوان یک رویکرد استاندارد توسعه خط تولید، می‌توان تا حدی آن را در سطح سوم بلوغ دانست ولی از آنجایی که این شرکت هیچ برنامه ارزیابی کمی مدونی برای سنجش خط تولید خود ندارد، قطعاً این شرکت در سطح چهارم بلوغ نخواهد بود.

## ۶-۴- ارزیابی ربع معماری

پس از ارزیابی خط تولید شرکت اعوان در بعد معماری بر اساس چارچوب FEF، به این نتیجه رسیدیم که این شرکت از جنبه‌ی «استفاده‌ی مجدد» در خط تولید به سطح ۵ رسیده است، چرا که تغییر سازوکار پیکربندی در این خط تولید میسر است، و به علاوه از تغییرپذیری در دارایی‌های نرم‌افزاری پشتیبانی می‌شود. البته این را هم باید ذکر کرد که بخشی از استفاده‌ی مجدد در خط تولید این شرکت مبتنی بر محصولات متن‌باز دیگران<sup>۳۹</sup> است.

با این حال، بررسی‌ها و مصاحبه‌های انجام‌شده نشان داد که شرکت برای خط تولید خود، معماری مرجع تعریف نکرده است، بلکه با توجه به توضیحات ارائه‌شده، در مورد هر محصول، معمار با توجه به نیازمندی‌های مشتری تصمیم می‌گیرد که چه اجزا و مؤلفه‌هایی را باید برای طراحی معماری آن محصول در نظر بگیرد، و آن‌گاه معماری این مؤلفه‌ها (در واقع همان دارایی‌های نرم‌افزاری) با توجه به مقتضیات مناسب‌سازی و پیکربندی می‌شود. روشن است که چنین رویکردی به کلی با مفهوم معماری مرجع متفاوت است، و بنابراین از جنبه‌ی «معماری مرجع» نمی‌توان خط تولید شرکت را در سطحی بالاتر از سطح ۱ ارزیابی کرد.

با توجه به آن‌چه گفتیم، نتیجه‌ی ارزیابی از جنبه‌ی «مدیریت تغییرپذیری» نیز نمی‌تواند سطحی بالاتر از سطح ۲ برای خط تولید این شرکت باشد؛ چرا که بعد از آن، ابتدا بر معماری مرجع، نقش مسلمی در ارتقا به سطوح بالاتر دارد. به این ترتیب با توجه به تفاوت و تباین آشکار مفاهیم و فرضیات چارچوب ارزیابی FEF و رویکردهای شرکت اعوان در بعد معماری، حتی چه بسا سطحی بالاتر از سطح ۱ را نتوان به عنوان نتیجه‌ی ارزیابی از این جنبه اعلام کرد.

به این ترتیب نتایج حاصل‌شده در بعد معماری ممکن است در نگاه اول چندان مطلوب نباشد، اما چنین به نظر می‌رسد که علت اصلی آن، اختلاف دیدگاه است. چارچوب ارزیابی FEF شرایطی را برای ارتقا به سطوح بالاتر در نظر می‌گیرد که ممکن است در عمل چندان مورد توجه قرار نگیرند، در حالی که شرکت یادشده به صورت تجربی به این رویکرد رسیده است. ضمن این که شرکت اعوان خط تولیدی برای خط تولید طراحی کرده است، و در واقع به مشتریان خود خط تولید عرضه می‌کند، و نه محصولات منفرد را. این دیدگاه را برخی پژوهش‌ها نیز مورد بررسی قرار داده‌اند (به ویژه یکی از آن‌ها که براساس واقعیت‌های تولید انجام شده است) و ما در ادامه بر همین موضوع تمرکز خواهیم کرد.

<sup>39</sup> third-party

## ۶-۵- ارزیابی ربع فرآیند

### من و خط تولید: فرآیند کجاست؟

فرآیند یکی از ارکان چهارگانه FEF است که مورد بررسی قرار می‌گیرد. در این ربع، سه خط مشخصه زیر وجود دارد:

- ۱- Domain Engineering
- ۲- Application Engineering
- ۳- Collaboration

در ادامه به سراغ هر یک از خطوط مشخصه بالا می‌رویم و از روی سؤالاتی که بر اساس ماتریس وضعیتهای FEF استخراج شده است، به بررسی ربع فرآیند برای خط تولید شرکت اعوان می‌پردازیم. لازم به ذکر است به ازای هر خانه از ماتریس وضعیتهای FEF (در ربع فرآیند)، چندین سوال (و حداقل یک سوال) در نظر گرفته شده است. این سؤالات از روی توضیح مشروحاتی که برای هر خانه در مستندات درسی (که خود برگرفته از کتاب SPLE است) بود، استخراج شده است. لذا می‌توان در نظر داشت که استخراج این سؤالات سلیقه‌ای نبوده است و تا حد خوبی، تکرارپذیر است.

### Domain Engineering

هدف از خط تولید، بازبهره‌بری<sup>۴۰</sup> از نهاده‌های ایجاد نرم افزار است. برای رسیدن به این هدف لازم است که دامنه ایجاد نرم افزار را مشخص کنیم. این گونه می‌توان بر روی مجموعه‌ای از مصنوعات هدف تمرکز کرد و با سامان دادن دامنه (از طریق مدیریت تنوع و در قالب مشخص کردن Variability ها و Commonality ها) به راه حلی مقدر برای بازبهره‌بری رسید.

اما اطلاق لفظ مهندسی بر این فعالیت سبب می‌گردد تا متوجه باشیم که مانند هر فعالیت مهندسی دیگر، باید ملاحظات مهندسی را مد نظر داشت: مواردی مانند در نظر گرفتن کیفیت، بودجه، زمان، پیش‌بینی پذیری در حد قابل قبول، Validation, Verification.

یک حسن و خوبی FEF در این جاست که نه تنها مدیریت پیچیدگی بر روی دامنه را در معیارهای خود مورد بررسی قرار می‌دهد بلکه نسبت به لازمه‌های یک فعالیت مهندسی هم آگاه است. یعنی ملاحظات مهندسی \_مانند موارد گفته شده در بالا\_ را نیز در معیارهای خود لحاظ می‌کند و مورد بررسی قرار می‌دهد.

پس در هنگام مواجهه با معیارهای FEF \_که در قالب ماتریس وضعیت FEF در منابع وجود دارد\_ باید دقت داشته باشیم که برخی معیارها با برخی دیگر مشابهت جدی دارند. زیرا مربوط به بررسی ملاحظات کلی یک فعالیت مهندسی می‌شوند.

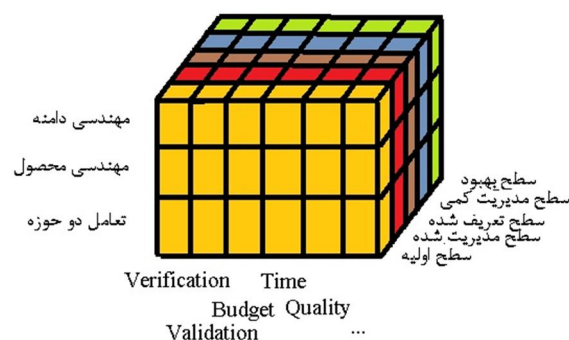
به عنوان نمونه، هم در خط مشخصه مهندسی دامنه به زیر معیار "بررسی وجود و سطوح Validation و Verification" بر می‌خوریم و هم در خط مشخصه مهندسی محصول. زیرا هر دو خط مشخصه یک فعالیت مهندسی هستند و باید این امر برای آنها مورد بررسی قرار بگیرد.

**پیشنهاد برای مشارکت علمی:** در اینجا است که ما به یک پیشنهاد برای بهبود و گسترش FEF دست می‌یابیم. همان طور که دیدیم مشترکاتی در بین معیارهای مربوط به خطوط مشخصه مختلف دیده می‌شود (مانند مثال بالا). چه خوب است خطوط

<sup>40</sup> reuseability

مشخصه به صورت یک فضای N- بعدی متعامد باشند. یک بعد همین خطوط مشخصه فعلی موجود در FEF است. به ازای هر Concern یک بعد جدید با خطوط مربوط به آن در نظر گرفته می شود. مثلاً برای "الزامات فعالیت‌های مهندسی" یک بعد در نظر گرفته می شود که دارای خطوط مشخصه ای مانند "Validation, Verification, کیفیت، بودجه، زمان و ..." است. حال به ازای هر نقطه در این فضای N-بعدی، یک تحلیل CMM صورت بگیرد. مثلاً بحث "Verification" در "مهندسی دامنه" خود دارای ۵ سطح است. همچنین بحث "Validation" در مهندسی دامنه دارای ۵ سطح است. هم چنین بحث Validation در مهندسی محصول دارای ۵ سطح است. بالطبع، بحث "Verification" برای مهندسی محصول دارای ۵ سطح خواهد بود. (هر ۴ جایگشت ممکن در این مثال دارای ۵ سطح بودند).

این گونه به چارچوبی Orthogonal برای بررسی خطوط تولید می رسیم که اصل symmetry را نیز رعایت کرده است. این اصول، (Orthogonal بودن، Symmetry, Zero-One-Infinity و ...) برای هر ساختمان منظم نرم که انسان با آن سر و کار دارد، مانند زبانهای برنامه سازی، مدل‌های مینا، معماری های نرم افزاری و چارچوب‌های نرم باید رعایت گردند.



شکل ۱۲- مثالی از یک چارچوب بررسی خط تولید پیشنهادی\_ البته مثالی و خیلی ساده\_ با ۲ بعد برای دو Concern مختلف و یک بعد هم برای تحلیل CMM

**Initial:**

۱- وجود Domain Activities در فرآیند مهندسی شرکت.

خلاصه وضعیت شرکت اعوان در این معیار: فعالیت‌های مهندسی دامنه در این شرکت در قالب پروژه WISE وجود دارد.

توضیحات کلی پیرامون این معیار: اگر خط تولید و ایجاد نرم افزار با رویکرد مبتنی بر مهندسی خانواده محصولات در یک شرکت وجود داشته باشد، باید اثری از فعالیت‌های مهندسی دامنه در آن شرکت دید. این فعالیتها مواردی مانند استخراج نیازمندی ها برای خانواده محصولات نرم افزاری، ساخت مصنوعات بازبهره‌رو برای اعضای خانواده و مواردی از این دست است. البته باید توجه داشت که فعالیت‌های مهندسی دامنه را می توان با دو رویکرد آگاهانه و غیرآگاهانه \_آنچنانکه ما آن را دریافتیم و می نامیم\_ انجام داد.

در رویکرد آگاهانه، ایجادکنندگان به تفاوت دامنه و محصول کاملاً آگاه هستند و فعالیتهای خود را در دو حوزه دامنه و محصول پی میگیرند. اما در رویکرد غیر آگاهانه، ایجادکنندگان کل خط تولید را یک "محصول نرم افزاری" جامع، زایا، کامل، قوی و با صفاتی از این دست می بینند. از این رو در حین ایجاد و توسعه این ابر محصول، قهرراً فعالیتهای مهندسی دامنه را نیز انجام می دهند: مثلاً سعی می کنند مجموعه ای کامل از نیازمندی ها بیابند (=به نوعی پرداختن به Commonality ها)، روشی برای پیکربندی و تنظیم نیازمندی ها برای مشتری های مختلف این ابر محصول در آن تعبیه می کنند(= نوعی از پرداختن به Variability ها) و ... ولی باید توجه داشت که رویکرد غیرآگاهانه از همگی مزایای خط تولید بهره مند نیست و عملاً می توان آن را حد واسطی بین سطح Initial یک خط تولید و عدم وجود خط تولید دانست: سطح زیر صفر.

البته باید به این نکته هم توجه داشت که سطح رویکرد غیرآگاهانه محل مناسبی برای رشد قابلیت یک شرکت برای تبدیل شدن به یک مجموعه خط تولید نرم افزار است. زیرا معمولاً رویکرد غیرآگاهانه به خط تولید ناشی از این است که سازمان یا شرکت مربوطه لزوم وجود چیزی مشابه خط تولید را دریافته است لیکن مصداق مناسب بیرونی آن را هنوز نیافته و مستقر نکرده است. از این رو چنین شرکتها و سازمانهایی می توانند با در اختیار گرفتن دانش مرتبط با خطوط تولید و بهره بردن از تجارب پیشینی دیگران، به رشدی سریع در این حوزه دست یابند: هم انگیزه سازمانی وجود دارد، هم پشتیبانی مدیریت و اعضای سازمان دیده می شود، هم نیاز و بازار مصرف برای این تلاش تجاری آنان از قبل موجود است. به تشبیه، سازمانها و شرکتهایی که در سطح رویکرد غیر آگاهانه به خط تولید قرار دارند، درد را احساس نموده ولی درمان را هنوز نیافته اند. چنین فردی زودتر از کسی که هنوز نمی داند نیاز به درمان دارد، علاج می یابد و جلوتر از او و به پیش رفته تر از اوست.

**توضیح پیرامون وضعیت شرکت اعوان در این معیار:** شرکت اعوان در ابتدای فعالیت خود\_سالیانی پیش تر\_ در سطح رویکرد غیرآگاهانه قرار داشته است. با احساس نیاز به خط تولید و وجود یک روند کلی تر برای ایجاد نرم افزار که به بازبهره‌روی<sup>۴۱</sup> و بازبهره‌بری<sup>۴۲</sup> و زایایی<sup>۴۳</sup> منجر شود، گام در مسیر بهبود وضعیت خود نهاد. یک تا دو محصول، به صورت خارج از رویکرد خط تولید ایجاد نمود، سپس وارد فاز رویکرد غیرآگاهانه به خط تولید گشت و پس از تکمیل اولین نسخه از خط تولید خود\_WISE\_ عملاً وارد سطح رویکرد آگاهانه به خط تولید گردید. نشانه این امر تمایز آگاهانه ای است که بین فعالیتهای قلمروی دامنه\_ در قالب پروژه WISE\_ و دیگر محصولات منبعث از WISE در این شرکت وجود داشته و دارد. لذا باید گفت شرکت اعوان یک سوژه جذاب پژوهشی است. این شرکت تمامی دوره های گذار یک شرکت از زمانی که خارج از گفتمان خط تولید به ایجاد نرم افزار می پردازد تا دوره بلوغ سطح پنجم\_چنانکه در صفحات ادامه این بررسی نتیجه خواهیم گرفت\_ را از سر گذرانده است. تحلیل ما این است که علت این رشد، همان احساس نیاز است. احساس نیاز به انگیزه و آگاهی سازمانی می انجامد و انگیزه سازمانی، پشتیبانی مدیریت و برنامه را از برنامه رشد به همراه خواهد داشت. آنچه نیاز باشد و خواسته شود و برایش هزینه گردد، به دست خواهد آمد.

**پیشنهاد برای مشارکت علمی:** از این فرصت می توان استفاده کرد و نتیجه گرفت که اولین فعالیت و شاید موثرترین فعالیت برای رشد یک سازمان یا شرکت در حوزه خط تولید، ایجاد یا کشف نیاز به خط تولید برای آن سازمان و شرکت

<sup>41</sup> Reusability

<sup>42</sup> Reuse

<sup>43</sup> Generation



است. نظر به این بررسی و دیگر بررسی های موجود در ادبیات این حوزه، می توان این امر را یک "الگوی فرایندی"<sup>۴۴</sup> و قطعه متد در این حوزه دانست.

**Managed.**

۲- مستندات پیرامون نیازمندی های خط تولید وجود دارد؟ (Requirement Engineering)

خلاصه وضعیت شرکت اعوان در این معیار: بله، وجود دارد.

توضیحات کلی پیرامون این معیار: مستندسازی ابزاری برای رسمیت بخشیدن و بازبهره‌بری از تصمیمات و تلاشی است که برای رسیدن به آن تصمیم صورت گرفته است. از این رو شرکتی که با عنایت به این دو وجه و دلیل مستندسازی، اقدام به تهیه مستندات بنماید، از فواید آن بهره‌مند خواهد شد. پس نه تنها باید مستندسازی نیازمندی ها مورد توجه قرار بگیرد، بلکه باید مستندات جایگاه رسمی و به منظور بهره برداری مجدد، داشته باشند.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان دارای مستندات نیازمندی های خط تولید می باشد. این مستندات هم تا حدودی برای شرکت رسمی به حساب می آیند و هم تا حدودی مورد استفاده مجدد قرار می گیرند.

۳- آیا در این مستندات، به تنوع و Variability هم به نحوی پرداخته شده است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، پرداخته شده است.

توضیحات کلی پیرامون این معیار: مساله اصلی که برای نیل به استفاده مجدد از مصنوعات باید حل گردد، مساله مدیریت تنوع و مدیریت گوناگونی مصنوعات است. از این رو مستنداتی در فرآیند خط تولید کارا و موثر خواهند بود که به این مساله پرداخته باشند.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان \_ تا حدی \_ رویکرد همه یا هیچ به نوع مصنوعات دارد. یعنی اگر تصمیم داشته باشد که Feature ای از دامنه را در قلمروی خط تولید ببیند و آن را در قالب مصنوعات دامنه محقق کند، سعی می کند تا انواع و ویژگی های متعدد آن Feature را شناسایی نماید. در این جاست که به طبقه بندی از تنوع موجود حول آن Feature دست می یابد که معادل یک Variation Point است. سپس بخشی از تنوع کشف شده را که صرفه اقتصادی دارد هدف گرفته و در قالب مصنوعات دامنه ( Code Generator هایی برای آن، کتابخانه هایی برای آن و ... ) محقق می سازد. رویکرد اصلی این شرکت در مدیریت تنوع، مدیریت تنوع از طریق زایایی<sup>۴۵</sup> است و نه استفاده مجدد صرف و کتابخانه‌ای. مثلاً به جای آن که ۱۰۰ فرم مینا داشته باشند، یک Form Generator را توسعه می دهند. البته از کتابخانه و استفاده مجدد صرف هم بهره مند هستند.

<sup>44</sup> Process Pattern

<sup>45</sup> Generation

۴- آیا فعالیتهای مربوط به **Plan** در **Domain** نمود و تاثیر دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، وجود دارد.

توضیحات کلی پیرامون این معیار: زمان و بودجه دو رکن از ارکان فعالیت مهندسی هستند که یکی از ابزارهای مهندسی برای در نظر داشت و نظارت بر آنها، **Plan** است. فعالیتهای مربوط به دامنه معمولاً زمان بر و هزینه بر هستند و از سوی دیگر در کوتاه مدت \_شاید\_ به بازدهی نرسند. لذا باید در **Plan** دیده شوند تا در زمان مناسب و پس از صرف هزینه لازم، به بهره‌برداری و ثمردهی برسند.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان از این نظر بسیار زیرکانه عمل نموده است. از سویی توسعه و تکمیل خط تولید خود را به صورت یک پروژه بلند مدت مدنظر دارد و به تکمیل آن می پردازد و از سوی دیگر، به ازای هر پروژه کوتاه مدت دیگر، سعی می کند تا ارزش افزوده ای برای خط تولید خود (در قالب مصنوعات جدید، تغییرات و ...) فراهم آورد. این گونه در تمامی **Plan** های پروژه های این شرکت، فعالیتهای قلمروی دامنه به خوبی مشخص و معلوم هستند و جایگاه دارند.

پیشنهاد برای مشارکت علمی: به نظر می رسد شرکت اعوان از یک رویکرد دورگه، تکراری و چرخه‌ای برای تکمیل خط تولید خود استفاده نموده است. رویکردی که منجر به ارائه چندین ویرایش از خط تولید و پیمودن مسیری تکاملی توسط آن شده است. این رویکرد، اکنون در آن حد پخته شده است که به عنوان یک فرآیند پیشنهادی برای شرکتهای داخلی و خارجی برای ایجاد خط تولید مورد استفاده قرار بگیرد. البته شاید نتوان یک فرآیند تمام عیار از آن استخراج نمود ولی می‌توان قطعات فرآیندی و الگوهای فرآیندی از آن استخراج نمود که در عمل در فضای مهندسی نرم افزار ما، کارا و موثر بوده اند.

۵- آیا ارتباط بین **Domain** و **Application** در **Plan** نمود و تاثیر دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، کاملاً وجود دارد.

توضیحات کلی پیرامون این معیار: دامنه و محصول و فعالیتهای مربوط به آن دو باید در یک مجموعه هماهنگ با یکدیگر به پیش بروند تا هم‌افزایی در فضای سیستمی آنها محقق گردد. از سوی دیگر، بخش مهمی از ارزش افزوده در پروژه ها، بستگی به مدیریت صحیح زمان و بودجه دارد که از طریق **Plan** صورت می گیرد. از این رو، یکی از مواردی که باید فعالیتهای دامنه و محصول در آن هماهنگ باشند، **Plan** است. گاهی یک فعالیت قلمروی دامنه متوقف و منوط به انجام یک فعالیت قلمروی محصول است. (مثلاً برای اخذ بازخوردهای فنی، آزمون و ...) هم چنین گاهی یک فعالیت قلمروی محصول متوقف و منوط به یک فعالیت قلمروی دامنه است. (مثلاً ایجاد و نهایی کردن تصمیمات معماری و طراحی پیرامون مصنوعات که قرار است مورد استفاده مجدد قرار بگیرند، مانند معماری مبنا، طرح آزمون و ...).

توضیح پیرامون وضعیت شرکت اعوان در این معیار: در شرکت اعوان **Plan** ها نه تنها به گونه ای ریخته می شوند که فعالیتهای دامنه و محصول در هماهنگی و هم‌افزایی انجام پذیرند، بلکه سعی می‌گردد که رویکردی چرخه‌ای و تکراری\_ولو غیر صریح\_ برای تکمیل و تغییر متوازی و متوازن دامنه و محصول وجود داشته باشد: هم دامنه، هم محصول، با هم در **Plan** مورد برنامه ریزی و تخصیص منابع قرار می‌گیرند.

۶- آیا استفاده از مصنوعات دامنه مورد نظارت قرار می‌گیرد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، از طریق فرآیند دورگه<sup>۴۶</sup> تیمی

توضیحات کلی پیرامون این معیار: تیم، شرکت یا سازمان باید مطمئن باشد که در مهندسی محصول و در جایی که امکان دارد، از مصنوعات دامنه استفاده مجدد صورت می‌گیرد. این گونه می‌تواند امید داشته باشد که از طریق استفاده مجدد، هزینه‌ای که برای مصنوعات دامنه کرده است باز گشته و پروژه خط تولید، به سود دهی می‌رسد.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان از نظر مدیریت منابع انسانی، رویکردی تیمی و چابک دارد. از این رو دارای انسجام بالایی در سازمان خود می‌باشد و عملاً به دلیل سایز کوچک سازمانی \_ در حدود ۲۰ نفر\_ بیشتر یک تیم است تا یک سازمان. از این رو نظارت بر کلیه امور \_ که یکی از آنها استفاده از مصنوعات دامنه است \_ از طریق فرآیندهای تیمی و چابک صورت می‌گیرد. مواردی مانند Face to Face Information Osmosis, Contacts. البته نمودهایی از فعالیتهای رسمی تر و فرآیندهای کلاسیک و سنگین تر هم در فعالیت این مجموعه دیده می‌شود. مانند مستند سازی ارتباطات درون سازمانی و بین افراد. البته همین مستندسازی هم از طریق روشی چابک \_ از طریق Forum اینترنتی شرکت \_ صورت می‌گیرد. رویکرد دورگه از مشخصات فرآیندی این شرکت است که ثمرات خوبی را برای آن به همراه داشته است.

۷- آیا دیدی کلی از وضعیت خط تولید موجود است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، در قالب نمودارهای UML.

توضیحات کلی پیرامون این معیار: وجود دید کلی از مصنوعات از دو جنبه حائز اهمیت است. اول آن که خود مصنوعات و خط تولید \_ به عنوان یک نرم افزار و ساخت نرم \_ محل تغییر و تحول و ایجاد هستند. از این رو لازم است که معماری و طرح کلی آنها موجود و در دسترس باشد تا از همه خوبی های معماری و طرح کلی بهره مند باشیم.

دوم آن که قرار است مصنوعات در مهندسی محصول مورد استفاده مجدد قرار بگیرند. از این رو باید بتوان نگاهی بین محصول و مصنوعات قلمروی دامنه برقرار نمود. وجود طرح کلی و دید کلی از وضعیت خط تولید و مصنوعات دامنه کمک می‌کند تا این نگاهت راحت تر برقرار شود. در واقع خط تولید بهتر فهمیده شود و مصنوعات مناسب با مهندسی محصول راحت تر و به نحوی موثر تر انتخاب و برگزیده شوند.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: در شرکت اعوان این امر در قالب متامدل های EMF و مدل های UML رخ داده است. از این مدلها هم برای درک خط تولید و هم توسعه و ایجاد ویرایش های بعدی آن استفاده می‌شود.

۸- آیا Versioning بر روی مصنوعات خط تولید اعمال می‌شود؟

<sup>۴۶</sup> نه آن چنان چابک که به بی قیدی منتهی شود و نه آنچنان کلاسیک و سنگین که عملی و مفید نباشد.

خلاصه وضعیت شرکت اعوان در این معیار: بله، نه تنها اعمال می شود بلکه به عنوان ابزاری در فرآیند ایجاد و مدیریت پروژه مورد توجه قرار می گیرد.

توضیحات کلی پیرامون این معیار: باید توجه داشت نه تنها باید Versioning روی مصنوعات وجود داشته باشد، بلکه باید از آن به عنوان ابزاری در فرآیند ایجاد و مدیریت پروژه \_چه در دامنه و چه در محصول\_ استفاده کرد. توضیح پیرامون وضعیت شرکت اعوان در این معیار: در شرکت اعوان به این امر کاملاً پرداخته شده است.

## Defined

۹- آیا Scope خط تولید تعریف شده است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، تعریف شده است.

توضیحات کلی پیرامون این معیار: همان طور که پیش تر هم گفته شد، خط تولید (یا در واقع خروجی مهندسی خانواده محصولات) خود یک ساختِ نرم و نرم افزار<sup>۴۷</sup> است و از این رو تمامی بایسته‌های یک نرم افزار یا فعالیت مهندسی نرم، برای آن هم بایسته و شایسته است. زین رو، همان طور که تعریف صریح Scope برای نرم افزار لازم است (تا انحراف از Scope<sup>۴۸</sup> و خزش موارد ناخواسته و نامربوط به Scope<sup>۴۹</sup> و دیگر مشکلات رخ ندهد).

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان به صورت مشخص و صریح قلمروی خط تولید WISE خود را مشخص کرده است: سیستم‌های نرم افزاری سازمانی Web-Based مبتنی بر تکنولوژی J2EE که عمده عملیات این سیستم‌ها، CRUD بر روی داده‌هاست. اگر دقت کنیم نه تنها Scope خط تولید این گونه مشخص شده است، بلکه تا حدودی مشخصات Platform هم تعیین گردیده است. یعنی تعریف Scope سطح بالای صرف نیست.

۱۰- آیا مشخصاً چند محصول در یک شاخه بازار هدف گذاری شده است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: این معیار یک پالایش و جلوه دقیقتری از معیار قبلی است.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، تمامی محصولات یک شاخه خاص و مشخص از بازار هدف قرار گرفته اند.

<sup>47</sup> (Nicolas Anquetil, 2010)

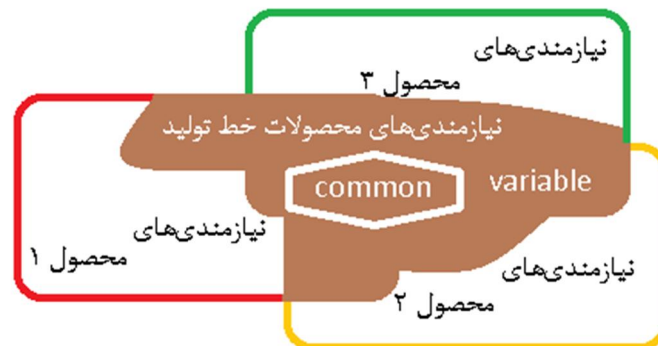
<sup>48</sup> Scope Drift

<sup>49</sup> Scope Creep

۱۱- آیا نیازمندیهای محصولات خط تولید، به نحوی تعریف شده است؟

### خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: در رویکرد خط تولید نه تنها بین دامنه و محصول تمایز وجود دارد بلکه بین خط تولید و محصولات نهایی نیز به عنوان ساختهای متفاوت نرم افزاری تمایز وجود دارد. همچنانکه پیشتر دیدیم که خود خط تولید دارای نیازمندیهای خاص خود است، محصولات نیز دارای نیازمندیهای خود هستند. حال بعضی از این نیازمندیها مشترکند و بعضی خاص یک محصول. در ضمن باید دقت داشت که برخی نیازمندیها اگرچه بین تمامی محصولات مشترک نیستند، اما ویژه یک محصول هم نیستند. در واقع نیازمندیهای غیرمشترک موجود در خط تولید هستند. ما می توانیم نیازمندیهایی داشته باشیم که در خط تولید دیده نشده اند و از طریق اجرای فعالیت Maintenance بر روی محصول پس از خروج از خط تولید بر روی آن اعمال شود. در این معیار، هدف این است که ببینیم آیا نیازمندیهای محصولات و نه خط تولید آن هم نیازمندیهایی که توسط خط تولید به نحوی تولید می گردند و نه لزوماً مشترک بین همه محصولات به نحوی تعریف شده اند.



شکل ۱۳- "نیازمندیهای محصولات خط تولید"، دقت شود که "نیازمندیهای خود خط تولید" به عنوان یک سیستم نرم افزاری باید جداگانه از "نیازمندیهای محصولات خط تولید" تحلیل و استخراج شود. البته "نیازمندیهای خود خط تولید" باید چنان باشد که بتواند محصولاتی با "نیازمندیهای محصولات خط تولید" ایجاد نماید.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، تعریف شده است. نیازمندیهای محصولات خط تولید تعریف شده اند.

۱۲- آیا Commonality ها و Variability ها به صورت صریح تعریف شده اند؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، در قالب متامدلی که برای خط تولید تعریف گردیده است. توضیحات کلی پیرامون این معیار: این امر لازم است چون امکان مدیریت تنوع و گوناگونی در دامنه را به ما می دهد و در واقع یکی از مهمترین وجوه مهندسی دامنه است. توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، در قالب متامدلی که برای خط تولید تعریف گردیده است. زیرا از روی این متامدل می توان دریافت که کدام وجوه تغییرپذیرند و کدام وجوه ثابت و مشترک.

۱۳- آیا محصولات خط تولید، به نحوی تعریف شده اند؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، باز هم در قالب متامدلی که برای خط تولید تعریف گردیده است. توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، باز هم در قالب متامدلی که برای خط تولید تعریف گردیده است، زیرا از روی این متامدل می توان دریافت که دامنه دارای چه مشخصات و تعاریف\_ البته سطح بالایی\_ است.

۱۴- آیا بستر معماری لازم برای دامنه در قالب یک راهکار فنی، تعریف شده و موجود است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله توضیحات کلی پیرامون این معیار: اگر بستر معماری لازم برای دامنه به شکل یک راهکار فنی (در واقع محیط پیاده سازی و بستر کد) موجود نباشد، فرآیند ایجاد عملاً ناهموار خواهد بود و این ناهمواری و نبود خودکار سازی، مانع از نموده شدن فواید خط تولید\_ خصوصاً باز بهره بری<sup>۵۰</sup> از مصنوعات\_ می شود. توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، از طریق تعریف افزونه هایی بر روی Eclipse و افزونه هایی برای EMF و همچنین سفارشی سازی خود Eclipse. همچنین از طریق تعریف معماری و ساختار استاندارد برای ساختار کد و کتابخانه ها.

۱۵- آیا Variability ها در راهکار فنی ارائه شده برای بستر معماری، نمود دارند؟ (وجود Interface های مختلف

تعریف شده و ...)

خلاصه وضعیت شرکت اعوان در این معیار: بله توضیحات کلی پیرامون این معیار: مشابه آنچه برای معیار قبلی گفته شد. توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، از طریق زبان استاندارد دی که برای هر افزونه تعریف گردیده است. در واقع رویکرد Model Driven شرکت اعوان سبب تسهیل رسیدن به خواست این معیار شده است.

۱۶- آیا Verification برای اطمینان از این که از مصنوعات دامنه در جای درست و به منظور صحیح استفاده شده

است صورت می گیرد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، البته مبتنی بر فرآیند تیمی شرکت توضیحات کلی پیرامون این معیار: مشابه آنچه در نرم افزار، Verification را الزامی می کند، هم در خط تولید\_ که یک سیستم نرم افزاری است\_ و هم در فرآیند ایجاد، باید این Verification موجود و ملموس باشد. در غیر این صورت نمی توان اطمینان نسبی برای رسیدن به سطح کیفیت مطلوب در محصولات یا خود خط تولید را داشت.

<sup>50</sup> Reuse

توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، از طریق فرآیند تیمی و انسجام بالایی که از نظر تیمی دارند، می‌توانند به سطح خوبی از Verification برسند. این گونه که هم برخی بررسی‌های رسمی را خصوصاً در پایان فرآیند ایجاد دارند و هم خود اعضای تیم کاملاً در این زمینه هماهنگ عمل می‌کنند. در واقع Collaboration تیمی برای رسیدن به سطح مطلوبی از Verification در حین فرآیند ایجاد، یک روش چابک است که توسط شرکت اعوان پیگیری می‌شود. اعضا به صورت غیرمتمرکز هم بر روی نتایج کار یکدیگر در فرآیند ایجاد عمل صحت‌سنجی را انجام می‌دهند و نتیجه را در صورت لزوم به یکدیگر گوشزد می‌کنند. البته خود شرکت اعوان هم با ما هم نظر است که از نظر Verification و آزمون باید فعالیت‌های بیشتر و موثرتری صورت دهد.

۱۷- آیا Validation بر روی مصنوعات دامنه برای اطمینان از این که معتبر و Valid هستند صورت می‌گیرد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: در رویکرد خط تولید، به صورت همزمان بر روی یک قلمروی نسبتاً متکثر که دامنه نامیده می‌شود کار صورت می‌گیرد. این خود چالشی برای Validation است. زیرا هر کدام از مصنوعات موجود در دامنه\_ولو یک طرح آزمون یا یک نمودار نیازمندی‌ها\_ باید Valid باشد چون به عنوان یک مصنوع و خروجی دامنه، قرار است بعداً مورد استفاده مجدد قرار بگیرد.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، تکیه شرکت اعوان بر نرخ بالای استفاده از مصنوعات است. این نرخ بالای استفاده از مصنوعات، اجازه نمی‌دهد که یک Artifact که غیر Valid است در مجموعه خط تولید باقی بماند و زود خود را نشان می‌دهد.

۱۸- آیا فرآیند سازمانی خط تولید تعریف شده و مستند است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: فرآیند برای یک فعالیت ایجاد، به مثابه سیستم عامل برای یک ماشین پردازنده است. فرآیند تعریف نشده، ناقص، معیوب، ناکارا، مبهم و کند می‌تواند تمام تلاش‌های صورت گرفته توسط بقیه اجزای سیستم و ارزش‌های تولید شده توسط آنان را خنثی و بی‌فایده کند.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، باز هم باید بر ذات دورگه و مهندسی شده فرآیند در شرکت اعوان تاکید کرد و آن را ستود. هم جوهی از فرآیندهای سنگین و کلاسیک\_ آنجا که لازم تشخیص داده شده است\_ دیده می‌شود (مانند مستندات تولیدی در فرآیند ایجاد) و هم چابکی فرآیند محسوس و مشخص است. حتی برخی تکنیکها و حتی فعالیت‌های ایجاد نرم افزار به نحو چابک در این شرکت مشاهده می‌گردد. (مانند نحوه Collaboration و مستندسازی ارتباطات سازمانی درون شرکت). البته میزان مستندات هنوز در تناسب با گستره خط تولید این شرکت کم است.

در شرکت اعوان تلاش شده است تا با بهره‌گیری از Model Driven Development و ابزارهای آن، مانند EMF، ایجاد مستندات و بروز نگه داشته شدن آنها تضمین شود. مثلاً فرض کنید اگر "مستند توصیف واسط

کاربر " ایجاد نگردد، پروژه Build یا Compile نخواهد شد. چون به نحو زایا و Model Driven، قرار است فرمهای UI از روی این مستند تولید گردد. همچنین این مستند بروز است چون تغییر در UI از طریق این مستند ممکن است و به نحو دیگر نمی توان تغییری وسیع و ساختاری در UI داد. لذا می توان مطمئن بود که مستند "توصیف واسط کاربر" همواره موجود و به روز است. چنین وضعی برای دیگر مستندات در شرکت اعوان\_ که در زنجیره ایجاد به نحو Model Driven وارد شده است\_ دیده می شود. البته هنوز مستندات مربوط به موارد کاربرد وارد این زنجیره نشده اند و وارد کردن آنها چندان آسان نمی نماید.

**پیشنهاد برای مشارکت علمی:** رویکرد شرکت اعوان برای تضمین در دسترس بودن مستندات و بروز بودن آنها می- تواند به عنوان یک "الگوی فرآیندی" و قطعه متد شناسایی شود. البته بعید به نظر می رسد که این الگو در ادبیات مربوط به MDD ناشناخته باشد ولی نکته با اهمیت، گزارش سازگار نمودن آن با رویکرد خط تولید در قالب یک تجربه موفق است.

۱۹- آیا فرآیند آموزش برای محصولات نهایی یا استفاده از مصنوعات دامنه تعریف شده است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: وجود فرآیند آموزش برای محصولات نهایی در رویکرد خط تولید اهمیت مضاعف دارد. زیرا مصنوعات دامنه متکثر و زیاد هستند و اگر مدیریتی برای تولید راهنمای مناسب برای آنها صورت نگیرد، در انتها ممکن است یا زمان و هزینه زیادی برای راه اندازی فرآیند آموزش مستقل مورد نیاز باشد یا فرآیند آموزش برای محصول نهایی غیر مفید و ناکارا باشد.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: رویه شرکت اعوان این گونه است که سعی می کند در کنار ایجاد هر کدام از مصنوعات دامنه، مستندات و مواد آموزشی لازم را هم تولید کند. هم چنین در مهندسی محصول، اقدام به یکپارچه سازی و سفارشی سازی و تکمیل این مواد آموزشی می کند.

## Quantitatively Managed

۲۰- آیا QPM وجود دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، تا حدودی

توضیحات کلی پیرامون این معیار: مدیریت کمی بر فرآیند کمک می کند که فرآیندها پیش بینی پذیرتر و مقیاس پذیر شوند، ریسکها سریع تر و قطعی تر شناسایی شوند، تجارب به نحوی هدفمند، نظام مند و تعریف شده شناسایی شده و مستند گردند. این گونه می توان امید داشت که در آینده قابلیت فعلی سازمان حفظ می شود و بستری برای بهبود مستمر\_ که سطح بعدی است\_ فراهم می آید و آماده می گردد.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: در شرکت اعوان، این امر بیشتر از طریق ابزارهای نرم افزاری موجود برای مدیریت فرآیند خصوصاً نرم افزاری های Issue Tracker\_ مشخصاً نرم افزار JIRA\_ تامین می گردد. البته در این زمینه شرکت اعوان جای کار دارد و می تواند با صرف توان، خود را توانا تر سازد.



۲۱- آیا معیارهای مدرج در قالب ابزار یا فرآیند منظم اندازه گیری یا ارزیابی می شوند؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: مشابه آنچه برای معیار قبلی گفته شد.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: نرم افزار JIRA برای این امر مورد استفاده قرار می گیرد.

۲۲- آیا بر روی نتایج آماری حاصل، تحلیل با روش مشخص (فرآیندی) صورت می گیرد؟

خلاصه وضعیت شرکت اعوان در این معیار: گاهی بله، گاهی خیر

توضیحات کلی پیرامون این معیار: برخی از ثمرات جمع‌آوری داده‌ها بلافاصله پس از مشاهده آنها بدست نمی آید و باید مورد تجزیه و تحلیل قرار بگیرند. مانند روند بازار، روندهای زمانی در فرآیند ایجاد، نرخ‌های وقوع و کشف خطا و ...

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان از حیث این معیار کمی ضعف دارد. البته به دلیل این که سائز شرکت در اشل‌های سازمانی چندان بزرگ نیست \_حدود ۲۰ نفر\_ لذا شاید نیاز و لزوم این امر احساس نشده است. البته هم اکنون هم گاهی از موارد \_مانند جمع‌بندی‌های دوره‌ای که بر روی خط تولید صورت می‌گیرد\_ این تجزیه و تحلیل‌ها صورت می‌گیرد ولی به صورت مستمر این امر رخ نمی‌دهد.

## Optimizing

۲۳- آیا بهبود هدف گذاری شده، برای مساله‌ای که از روی معیارهای عمیقاً تعریف شده کشف یا مشخص گردیده است،

وجود داشته است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: بهبودی در این سطح مدنظر است که به صورت آگاهانه، هدفمند و مورد انتظار بدست آمده باشد. بهبودهای تصادفی، غیرآگاهانه و یا غیرمترقبه نشان از بلوغ یک سازمان نیست.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان تا کنون ۴ ویرایش از خط تولید خود را تولید کرده است و با هر ویرایش چندین محصول تولید گشته و از خط تولید خارج شده است. این روند هم اکنون هم ادامه دارد و مستمر است. این ویرایش‌ها به صورت هدفمند و به منظور ارتقای قابلیت‌های خط تولید یا کیفیت آن تولید می‌گردند و نشان از تحقق بهبود مستمر هستند. پس از نظر این شاخص وضع شرکت اعوان مطلوب است.

## Application Engineering

.Initial

۱- وجود فعالیتهای مهندسی محصول

خلاصه وضعیت شرکت اعوان در این معیار: بله، وجود دارد.

توضیحات کلی پیرامون این معیار: فعالیتهایی مانند استخراج نیازمندی‌ها، تحلیل و طراحی مهندسی و فعالیتهای چتری.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: در وضع کاملا مطلوب وجود دارد و تا حدی برگرفته از متدولوژی RUP است. البته همان طور که قبلا هم گفتیم فرآیند این شرکت مهندسی شده و نمی توان آن را کاملا کلاسیک یا کاملا چابک دانست.

**Managed:**

۲- آیا نیازمندی‌ها استخراج می شوند؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: یکی از پایه‌ای ترین و با اهمیت ترین فعالیتهای مهندسی محصول است. توضیح پیرامون وضعیت شرکت اعوان در این معیار: مطلوب است. در قالب موارد کاربرد و به صورت تفصیلی و مفصل استخراج می شود.

۳- هم به صورت مستقل هم به صورت باز استفاده از نیازمندیهای دامنه؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: برای این که مدیریت تنوع ممکن باشد، لازم است تا نیازمندی های محصول به دامنه نگاشته شوند.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: در هر پروژه و برای هر محصول، به صورت مشخص در فعالیتی تعریف شده، سعی می شود که نیازمندی‌های محصول به دامنه نگاشته شوند و در صورت امکان، از نیازمندی‌های دامنه، استفاده مجدد صورت گیرد. نام این فعالیت در شرکت اعوان، "گونه‌شناسی" است.

۴- آیا در Plan استفاده از مصنوعات دامنه دیده شده است یا تاثیری صریح در آن دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: مشابه آنچه در بخش مربوط به مهندسی دامنه ذیل معیار مشابه توضیح داده شد.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: مشابه آنچه در بخش مربوط به مهندسی دامنه ذیل معیار مشابه توضیح داده شد.

۵- آیا تحلیل ریسک وابستگی به فعالیتهای دامنه بر روی Plan انجام می شود؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: این وابستگی یک نیاز عینی در مهندسی خانواده محصولات است تا بتوان به استفاده مجدد از مصنوعات دست یازید.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، در Plan کاملاً این وابستگی آگاهانه مدیریت می گردد.

۶- آیا تاثیر مهندسی دامنه بر Scope پروژه ها وجود دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: این امر هم اجتناب ناپذیر است. زیرا وقتی نیازمندیها مرتبط و از روی دامنه تعریف می شوند، Scope هم باید تا حد زیادی با دامنه منطبق و مشترک باشد.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، کاملاً وجود دارد. حتی بعضی از پروژه های پیشنهادی به شرکت صرفاً به دلیل عدم انطباق این Scope ها اجرایی نمی گردند و شرکت اعوان آنها را نمی پذیرد.

۷- آیا استفاده از مصنوعات دامنه در مهندسی محصول تحت Measurement یا تحلیل قرار دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، ولی به صورت محدود

توضیحات کلی پیرامون این معیار: بحث Measurement برای استفاده از مصنوعات دامنه زمانی معنی بیشتر و دقیقتری دارد که در سازمانی با ابعاد بزرگ مساله را پیگیری نماییم وگرنه در تیم های ایجاد که به صورت هماهنگ و منسجم فعالیت می نمایند این بحث چندان معنایی ندارد. البته تحلیل در هر دو جا مهم است.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: در شرکت اعوان بر روی استفاده از مصنوعات دامنه تحلیل صورت می گیرد.

۸- آیا بازاستفاده از محصولات دامنه تاثیر خود را در Versioning محصول می گذارد؟ یا مبنایی برای Versioning

محصول هست؟ یا تحت Versioning قرار می گیرند؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: شماره گذاری ویرایشی و تعریف نسل، ابزاری برای مدیریت انواع پیچیدگی هایی است که ممکن است در هر قسمت از فرآیند ایجاد بروز و ظهور داشته باشد. (مانند پیچیدگی ایجاد، پیچیدگی آزمون، پیچیدگی صحت سنجی و واریسی نرم افزار، پیچیدگی های مدیریت پروژه و ...). حال خود این زیرفرآیند Versioning می تواند محل پیچیدگی باشد. پیچیدگی خود زیرفرآیند Versioning را می توان به روش های مختلفی مدیریت کرد (= یعنی نحوه تعریف و تعیین Version ها و ...). یک رویکرد، رویکرد بازگشتی است: یعنی Version کل نرم افزار از روی Version اجزای آن تعریف و تعیین گردد.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: در شرکت اعوان این امر وجود دارد و یکی از ملاکهای Versioning کلی است.

۹- برای یک محصول یا مشتری خاص، نیازمندی‌ها استخراج شوند. در حین استخراج نیازمندی‌ها، از نیازمندی‌های موجود در دامنه استفاده مجدد شود. همچنین نیازمندی‌های محصول به نیازمندی‌های دامنه مرتبط و متصل شوند. آیا چنین فرآیندی در مهندسی محصول تعریف شده و وجود دارد؟

#### خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: استفاده مجدد از مصنوعات و نیازمندی‌های دامنه، این امر را الزامی می‌کند. توضیح پیرامون وضعیت شرکت اعوان در این معیار: همان‌طور که قبلاً توضیح داده شد، در هر پروژه و برای هر محصول، به صورت مشخص در فعالیتی تعریف شده، سعی می‌شود که نیازمندی‌های محصول به دامنه نگاشته شوند و در صورت امکان، از نیازمندی‌های دامنه، استفاده مجدد صورت گیرد. نام این فعالیت در شرکت اعوان، "گونه شناسی" است.

۱۰- آیا از محیط نرم افزاری تعریف شده ای برای استفاده مجدد از مصنوعات دامنه در فرایند مهندسی محصول استفاده شده است؟

#### خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: هرگونه ابهام یا شکاف در زنجیره فرآیندها و ابزارها، موجب پیچیدگی آفرینی در فرآیند ایجاد شده که خود را در قالب خطاها، هزینه‌ها و دیگر موارد نامطلوب و سربار نشان می‌دهد. توضیح پیرامون وضعیت شرکت اعوان در این معیار: از محیط Eclipse\_ که به کمک افزونه‌هایی سفارشی سازی هم شده است\_ برای این منظور استفاده می‌شود.

۱۱- Verification و Validation در فرآیند ایجاد به نحوی تعریف شده هم بر روی مصنوعات دامنه، هم نسخه تنظیم شده آنها و هم مصنوعات ویژه محصول انجام می‌شود؟

#### خلاصه وضعیت شرکت اعوان در این معیار: بله، البته باید بهبود بیابد.

توضیحات کلی پیرامون این معیار: پیشتر در بخش مربوط به مهندسی دامنه در ذیل معیارهایی که مشابه با این معیار بوده‌اند، توضیح داده شده است. البته علاوه بر آن موارد باید دانست که بحث آزمون و واریسی، بحثی دامنه‌دار خصوصاً برای خطوط تولید است و هنوز در حیطه خط تولید، به ساحل آرامش و توافق جمعی نرسیده است. زیرا خطوط تولید سرباری از مرتبه ۳ برابر در هزینه‌ها به همراه دارند و خود آزمون و واریسی در سطوح عالی و کامل، سرباری ۳ برابری را به همراه دارد. از این رو یک خط تولید خوب و ایده آل، یک خط تولید مرده خواهد بود: چون سرباری ۹ برابری (۹۰ درصد هزینه تملک خط تولید به ازای هر واحد هزینه) را خواهد داشت که نرخ سرمایه‌گذاری بالایی است و در محدود پروژه‌هایی به صرفه خواهد بود.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان هم از رویه کلی موجود در خطوط تولید مستثنی نیست و باید در انتظار رشد بیشتر این شرکت از نظر توصیف رسمی، واریسی و آزمون نرم‌افزار باشیم. هرچند به نسبت در وضعیت خوبی به سر می‌برد.

۱۲- افراد فنی آموزش لازم برای استفاده مجدد از مصنوعات دامنه را دیده‌اند؟ آیا آگاه هستند که از هر مصنوع دامنه چه کاربردی می‌توان گرفت؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: وجود ابزاری بدون استفاده از آن فقط سربار است. برای استفاده، آموزش لازم است.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: این آموزش‌ها در شرکت اعوان وجود دارد. هم به صورت متمرکز در ابتدای پیوستن اعضا به تیم و هم به صورت مستمر. همچنین در هر پروژه، فردی به نام WISE Mentor وجود دارد که آگاه به خط تولید WISE است و دیگر اعضای تیم به مجرد احساس کمبود از نظر اطلاعات پیرامون WISE به نزد او می‌روند. همچنین فرآیندی برای مستندسازی ارتباطات اعضا از طریق Forum اینترنتی شرکت وجود دارد که موجب می‌شود تا مواد آموزشی لازم در هر زمان و به نحوی جست و جو پذیر در اختیار افراد باشد.

۱۳- آیا گروه‌های خاصی وجود دارد که آموزش دیده باشند که به صورت تیمی پیوند بین پروژه خاص و دامنه را برقرار کنند و محقق سازند؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: نهادسازی برای یک نیاز یا دغدغه، یک شیوه مدیریت آن نیاز است. این معیار هم بررسی می‌کند که برای دغدغه هماهنگی و ارتباط دامنه و محصول، نهادی در سازمان تحت بررسی وجود دارد یا خیر.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: معمار هر محصول از بین طراحان خط تولید انتخاب می‌شود. همچنین در هر پروژه، فردی به نام WISE Mentor وجود دارد که آگاه به خط تولید WISE است و مواظب این دغدغه است.

## Quantitatively Managed

۱- آیا QPM برای مهندسی محصول وجود دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، تا حدودی

توضیحات کلی پیرامون این معیار: مشابه توضیحی که در بخش معیار QPM در مهندسی دامنه داده شد.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: بله، البته از طریق ابزارهای Issue Tracker مشخصاً ابزار

JIRA

۲- آیا معیارهای مدرج در قالب ابزار یا فرآیند منظم اندازه گیری یا ارزیابی می شوند؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

۳- آیا بر روی نتایج آماری حاصل، تحلیل با روش مشخص (فرآیندی) صورت می گیرد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، البته به صورت دوره‌ای و گاه گاه و نه مستمر که بهتر است شکل مستمر به خود بگیرد.

## Optimizing

۱- آیا بهبود هدف گذاری شده، برای مساله ای که از روی معیارهای عمیقاً تعریف شده کشف یا مشخص گردیده است، وجود داشته است؟

مشابه آنچه در بخش قبلی ذکر شد:

خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: بهبودی در این سطح مدنظر است که به صورت آگاهانه، هدفمند و مورد انتظار بدست آمده باشد. بهبودهای تصادفی، غیرآگاهانه و یا غیرمترقبه نشان از بلوغ یک سازمان نیست.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان تا کنون ۴ ویرایش از خط تولید خود را تولید کرده است و با هر ویرایش چندین محصول تولید گشته و از خط تولید خارج شده است. این روند هم اکنون هم ادامه دارد و مستمر است. این ویرایش‌ها به صورت هدفمند و به منظور ارتقای قابلیت‌های خط تولید یا کیفیت آن تولید می گردند و نشان از تحقق بهبود مستمر هستند. پس از نظر این شاخص وضع شرکت اعوان مطلوب است.

## Collaboration

.Initial

۱- آیا فعالیتی برای ارتباط و همکاری بین دامنه و محصول وجود دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، در قالب Wise Mentor و حضور افراد در Role هایی هم در دامنه و هم محصول.

.Managed

۲- آیا ردگیری دوجهته بین نیازمندی های دامنه و محصول وجود دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

۳- آیا طول عمر بیشتر مصنوعات دامنه به نسبت محصول در Plan دیده شده است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

۴- آیا در Plan همگام کردن مهندسی دامنه و محصول دیده شده است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

۵- آیا نظارتی بر ارتباط بین دامنه و محصول هست؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

۶- آیا نظارت و کنترلی برای همگامی مهندسی دامنه و محصول انجام می شود؟ (مثلا تعریف نقاط همگامی)

خلاصه وضعیت شرکت اعوان در این معیار: بله

۷- آیا Versioning و Configuration Management در دامنه بر محصول یا بالعکس اثر دارد؟ آیا همگام

هستند؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، این امر در ویرایش های جدیدی که از خط تولید ارائه می گردد\_ که

تا به حال به ۴ مورد رسیده است\_ رعایت می گردد. همچنین هماهنگ بودن و پشتیبانی از محصولات ویرایش های

قبلی خط تولید هم مد نظر است. زیرا از WISE برای نگهداری محصولات تولید شده توسط WISE هم استفاده می

گردد و همچنین این امر\_ یعنی پشتیبانی از موارد پیشین\_ نیاز به آموزش مجدد را مگر در مواردی که لزوم دارد از

بین می برد.

Defined

۱- آیا فرآیند تعریف شده برای تکمیل احتمالی نیازمندی ها و دیگر مصنوعات دامنه از روی نیازمندی ها و مصنوعات محصولات وجود دارد؟

خلاصه وضعیت شرکت اعوان در این معیار: بله، همان طور که پیشتر هم گفته شد در قالب "گونه شناسی" ارتباط بین دامنه و محصول از نظر نیازمندی ها بررسی می شود و در این فعالیت و در دیگر فعالیتها، بازخوردهایی برای تکمیل نیازمندی های دامنه جمع آوری می گردد که در حین بروزرسانی های خط تولید مد نظر قرار می گیرد. ابزار Issue Tracker مورد استفاده یعنی JIRA نیز بستری برای پردازش با تاخیر بازخوردها فراهم آورده است. باید متذکر شد که فرآیند تکمیل و اصلاح خط تولید موازی با فرآیند جاری شرکت مد نظر است و گاهی هم تلاقی دارد.

۲- آیا روش و معیار تعریف شده برای انتخاب از بین مصنوعات و برای مصنوعات چه در دامنه و چه در مهندسی محصول وجود دارد؟ آیا این امر با ابزار و محیط ایجاد یکپارچه شده است؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

۳- آیا تصمیم سفارش و خرید، استفاده مجدد یا ساخت از نو، به صورت صریح برای مصنوعات، تصمیم سازی و تصمیم گیری می شود؟

خلاصه وضعیت شرکت اعوان در این معیار: بله

۴- وجود نصب العین و نقشه راه برای محصولاتی که در آینده قرار است به مجموعه خط تولید افزوده شود یا از طریق خط تولید، ایجاد گردند.

خلاصه وضعیت شرکت اعوان در این معیار: بله

۵- تعریف زمان بندی و توالی که برای تحویل و انتقال محصول و فازها به مشتری نیاز است.

خلاصه وضعیت شرکت اعوان در این معیار: بله

۶- امکان سازهمبندی<sup>۵۱</sup> مصنوعات دامنه و محصول

خلاصه وضعیت شرکت اعوان در این معیار: بله

۷- داشتن توامان و سازگار و هماهنگ Verification و Validation در دامنه و محصول

خلاصه وضعیت شرکت اعوان در این معیار: بله

۸- تبادل نتایج Verification و Validation بین دامنه و محصول

خلاصه وضعیت شرکت اعوان در این معیار: بله

<sup>51</sup> Integration



۹- به اشتراک گذاری یک سیاست Planning بین دامنه و محصول.

خلاصه وضعیت شرکت اعوان در این معیار: بله

۱۰- اهداف و اهداف اجرایی و سطح عملکردی مورد انتظار سازمان از خط تولید تعریف شده باشد.

خلاصه وضعیت شرکت اعوان در این معیار: بله، البته تعریف شده هستند اما به صورت رسمی بین تمامی اعضاء

شرکت منتشر نشده اند و مدیران ارشد شرکت از آن آگاه هستند.

۱۱- همگام سازی برنامه های عملیاتی بین دامنه و محصول.

خلاصه وضعیت شرکت اعوان در این معیار: بله

۱۲- مسئولیتهایی که بین چندین پروژه یا محصول مشترک باشد در سازمان تعریف شده باشد. در واقع در فرآیند

سازمانی، ارتباط بین دامنه و محصول در قالب نقش های سازمانی تعریف شده باشد.

خلاصه وضعیت شرکت اعوان در این معیار: بله، معماران پروژه ها از بین طراحان Wise انتخاب می شوند و هم چنین

یک نقش WISE Mentor هم در هر پروژه موجود است. پیشتر، بیشتر توضیح داده شده است.

۱۳- برقراری ارتباط و مدیریت بین مصنوعات و محصولات فعلی و قبلی و آتی در دامنه و محصولات

خلاصه وضعیت شرکت اعوان در این معیار: بله

۱۴- وجود فرآیند تعریف شده برای تشخیص مصنوعات جدید برای دامنه از روی محصولات

خلاصه وضعیت شرکت اعوان در این معیار: بله

۱۵- مدیریت ریسک و راهبرد آن هم پیرامون دامنه و هم محصول موجود و مناسب باشد.

خلاصه وضعیت شرکت اعوان در این معیار: بله

## Quantitatively Managed

۱۶- مدیریت QPM برای وابستگی های بین دامنه و محصول و نحوه پیشبرد و پیشرفت در فعالیتهای همگام سازی

خلاصه وضعیت شرکت اعوان در این معیار: بله

۱۷- برقراری ارتباط برای بازخورد بین دامنه و محصول

خلاصه وضعیت شرکت اعوان در این معیار: بله

۱۸- بررسی گلوگاه های مابین این دو

## خلاصه وضعیت شرکت اعوان در این معیار: بله

۱۹- راهبری و نظم بخشی به افراد در هر دو حوزه و با دید توامان

خلاصه وضعیت شرکت اعوان در این معیار: بله، همانطور که پیشتر گفته شد، معماران پروژه ها از بین طراحان Wise انتخاب می شوند و همچنین یک نقش WISE Mentor هم در هر پروژه موجود است.

## Optimizing

۱- آیا بهبود هدف گذاری شده، برای مساله ای که از روی معیارهای عمیقاً تعریف شده کشف یا مشخص گردیده است، وجود داشته است؟

مشابه آنچه در بخش قبلی و بخش قبل از آن هم ذکر شد:

## خلاصه وضعیت شرکت اعوان در این معیار: بله

توضیحات کلی پیرامون این معیار: بهبودی در این سطح مدنظر است که به صورت آگاهانه، هدفمند و مورد انتظار بدست آمده باشد. بهبودهای تصادفی، غیرآگاهانه و یا غیرمترقبه نشان از بلوغ یک سازمان نیست.

توضیح پیرامون وضعیت شرکت اعوان در این معیار: شرکت اعوان تا کنون ۴ ویرایش از خط تولید خود را تولید کرده است و با هر ویرایش چندین محصول تولید گشته و از خط تولید خارج شده است. این روند هم اکنون هم ادامه دارد و مستمر است. این ویرایشها به صورت هدفمند و به منظور ارتقای قابلیت‌های خط تولید یا کیفیت آن تولید می گردند و نشان از تحقق بهبود مستمر هستند. پس از نظر این شاخص وضع شرکت اعوان مطلوب است.

## جمع‌بندی کلی وضعیت رویکرد شرکت اعوان به خط تولید از نظر فرآیند

- این شرکت در سطح پنجم از بلوغ از نظر فرآیند در زمینه خط تولید قرار دارد. البته این نافی شکننده بودن وضعیت این شرکت در برخی از معیارها نیست. ذات تیمی و منسجم این شرکت \_ که یکی از دلایل آن سائیز سازمانی نسبتاً کوچک آن یعنی ۲۰ نفر است \_ به مجرد افزایش مقیاس شرکت مورد تهدید قرار می‌گیرد. پیش‌بینی ما این است که اگر مقیاس شرکت دو برابر گردد، و کار دیگری صورت نگیرد، سطح بلوغ شرکت از سطح ۵ به سطح ۳ افت خواهد کرد. زیرا برخی از معیارهای مهم موجود در بخش ۳ اکنون از طریق انسجام بالای تیمی و فرآیند شبه چابک موجود در شرکت تامین می‌گردد.
- مشخصاً نقاطی که باید برای ترمیم و ارتقای آنها توان صرف گردد و جای شکوفایی بیشتر دارد عبارت است از:
  - مستند سازی خصوصاً مستندسازی همگام با دیگر فعالیتها و نه به صورت دوره‌ای و آن هم به صورت قطعی و صریح و نه ضمنی و Collaborative ...
  - سنگین تر شدن و تعریف شده تر بودن فعالیت‌های مربوط به وارسی و آزمون
  - استفاده بیشتر و استفاده از تعداد بیشتری از معیارها و سنج‌ها (Measurements and Metrics) در فرآیند ایجاد نرم افزار
    - متنوع کردن ابزارهای مدیریت پروژه و عدم اتکای صرف بر JIRA
    - تعریف تعداد جلسات بیشتر و فعالیت‌های مرور منظم تر به منظور تحلیل داده‌ها و متریک‌ها
    - تعریف فرآیند و گام‌های مشخص برای شناخت تغییرات در فضای تکنولوژی و دانش روز: زیرا شرکت اعوان را یک شرکت دانش‌بنیان تشخیص داده ایم. شرکتی که تکیه‌اش و سرمایه‌اش، برش تکنولوژیک و دانشی است.
    - مثلاً تمامی اعضای شرکت موظف باشند در گروه‌های ایمیلی جهانی مرتبط با زمینه کاری خود عضو باشند و هر هفته گزارشی محدود (مثلاً یک جمله‌ای) از تغییرات به مدیریت ارشد و اتاق فکر شرکت ارائه دهند.

## ۶-۶- ارزیابی ربع سازمان

در شرکت اعوان نقش‌ها و مسئولیت‌ها کاملاً منطبق با الگوی خط تولید تعریف شده است. مثلاً نقش خاصی برای برقراری ارتباط میان core و سایر برنامه‌ها در هر پروژه تعریف شده است که wise mentor نام دارد. ضمناً ساختار سازمانی نیز در سطوح مختلفی مانند تحلیل، طراحی و پیاده‌سازی، الگوی wise را رعایت می‌کند. جلسات مناسبی هم برای انتقال تجربه میان تیم core و سایر تیم‌ها به صورت منظم برگزار می‌شود. بنابراین شرکت اعوان را می‌توان شرکتی با سطح بلوغ ۵ در زمینه سازمان‌های مبتنی بر خط تولید دانست.

## ۷- بررسی مقالات مرتبط با رویکرد اعوان در زمینه خط تولید

### ۷-۱- مقدمه

در این فصل، با مطالعه مقالات دانشگاهی منتشر شده پیرامون خط تولید که بیشترین تطابق را با مدل خط تولید شرکت اعوان دارند، یعنی رویکرد عمدتاً پایین به بالا، تلاش می‌شود تا با تحقیقات و پژوهش‌هایی که در مورد این دسته از شرکت‌ها صورت گرفته است آشنایی بیشتری فراهم شود.

### ۷-۲- تغییرپذیری یا ترکیب؟

طراحی خط تولید به طور کلی شامل این مراحل است: ابتدا مشخص کردن اشتراک‌ها<sup>۵۲</sup> و گوناگونی‌ها، سپس ساخت یک زیرساخت (یا معماری مرجع) از اشتراک‌ها فارغ از گوناگونی‌ها، و آن‌گاه تعیین نقاط تغییر<sup>۵۳</sup> و گونه‌ها<sup>۵۴</sup>. هدف از انجام این مراحل نیز مدیریت پیچیدگی تولید تعداد انبوه محصولات است. در عین حال در [1] این موضوع عنوان شده که هرچند اتخاذ چنین رویه‌ای برای خانواده‌های کوچک محصولات مناسب است، اتکا بر آن برای تولید گروه‌های بزرگ‌تری از محصولات (به ویژه در شرایطی که موضوع فراتر از مرزهای سازمان<sup>۵۵</sup> باشد) مشکلاتی را به وجود خواهد آورد.

به این ترتیب راه حلی فراگیر مورد نظر است که بتواند از عهده‌ی پیچیدگی و تنوع گسترده‌ی وسیعی از محصولات برآید؛ به بیان دیگر در شرایطی که محدوده‌ی خط تولید توسعه می‌یابد و شرکت به سمت تولید جمعیت محصولات<sup>۵۶</sup> به جای خانواده‌ی محصولات حرکت می‌کند، به ویژه وقتی تولید در سازمان‌های متفاوت یک شرکت انجام می‌شود، راه‌حل‌های سنتی خط تولید کم‌کم موجد مشکلاتی خواهند شد. به اعتقاد [1] ساخت جمعیت محصولات، چالش بعدی در مهندسی نرم‌افزار خواهد بود.

منبع اخیر، ۴ مورد را به عنوان انگیزه‌های روی آوردن به خط تولید برمی‌شمارد: مدیریت اندازه و پیچیدگی، دستیابی به کیفیت بالا، مدیریت تنوع<sup>۵۷</sup>، و کاهش زمان تولید. از این میان به ویژه به مدیریت تنوع توجه می‌کند و دو روش کلی را برای آن قابل تصور می‌داند که به نوعی دو سر یک طیف‌اند: اول به کارگیری یک معماری که اشتراک‌ها را دربرمی‌گیرد و نقاط تغییر را برای گوناگونی‌ها تعریف می‌کند، و دوم استفاده از مؤلفه‌های نرم‌افزاری برای پیاده‌کردن اشتراک‌ها و پشتیبانی از گوناگونی‌ها از طریق تعیین نحوه‌ی ترکیب

<sup>52</sup> commonality

<sup>53</sup> variation point

<sup>54</sup> variant

<sup>56</sup> product population

<sup>57</sup> diversity

<sup>55</sup> منظور از سازمان در این‌جا سازمانی در داخل یک شرکت بزرگ است.

این مؤلفه‌ها برای ساخت محصول. هدف این است که راه حل مطلوب، از تعادل این دو روش کلی با توجه به محدوده‌ی محصولات و ویژگی‌های سازمان، پیدا شود. البته این منبع رویکرد دیگری را نیز که مبتنی بر ارث‌بری و زیرکلاس‌سازی است برای بهره‌گیری از مزایای استفاده‌ی مجدد ممکن دانسته است.

### ۷-۳- تأثیر محدوده بر خط تولید

پیش‌تر هم گفته شد که روش سنتی طراحی و پیاده‌سازی خط تولید از تشخیص زودهنگام اشتراک‌ها و گوناگونی‌ها، و آن‌گاه تعیین یک معماری مرجع شامل تعدادی نقطه‌ی تغییر پشتیبانی می‌کند. با وجودی که این روش اصولاً درست است، اما در بعضی شرایط ممکن است چندان موفق عمل نکند. این شرایط از دو جنبه‌ی محصول محور و سازمان محور قابل بررسی است.

از جنبه‌ی محصول محور، روش سنتی اغلب، خانواده‌هایی از محصولات را پوشش می‌دهند که بیش‌تر اشتراک و کم‌تر گوناگونی دارند. به این ترتیب تعیین یک معماری خالی از گوناگونی‌ها (تنها دربرگیرنده‌ی اشتراک‌ها) که دارای تعدادی نقطه‌ی تغییر است از تنوع محصولات پشتیبانی می‌کند، ما را به مقصود می‌رساند. هرچند نقاط تغییر عمل کرد داخلی مؤلفه‌ها را کنترل می‌کنند، اما کاربرد یا عدم کاربرد مؤلفه‌ها، هم‌چنین انتخاب مؤلفه‌ها برای پیاده‌سازی مؤلفه‌های انتزاعی، و هم‌چنین اتصال آن‌ها نیز از مواردی است که ممکن است توسط مؤلفه‌ها کنترل شود (برخی از این موارد در نمودار متعامد<sup>۵۸</sup> قابل نمایش است).

البته در [1] این موضوع بیان شده که تمایل بازار به سمت «ترکیب» قابلیت محصولات گوناگون است؛ مثلاً گوشی‌های تلفن همراه که دارای دوربین یا پخش‌کننده‌ی موسیقی شده‌اند. این منبع معتقد است که در این شرایط باید به سمت معماری‌های ترکیبی رفت. به بیان دیگر، در حالی که تغییرپذیری هم‌چنان می‌تواند برای خانواده‌های محصولات که تنوع محدود و شناخته‌شده‌ای دارند گزینه‌ی مناسبی باشد، ترکیب گزینه‌ای که در شرایط محدوده‌ی ناشناخته‌ی محصولات می‌تواند مورد توجه ویژه قرار گیرد.

از سوی دیگر، از جنبه‌ی سازمان محور، شرکت‌های بزرگ (نظیر شرکت فیلیپس که در این منبع بررسی شده) اغلب به صورت در قالب بخش‌های مسوول محصولات<sup>۵۹</sup> سازمان‌دهی می‌شوند، و هر یک از این بخش‌ها نیز به نوبه‌ی خود به واحدهای کسب‌وکار، و آن‌ها نیز به خطوط کسب‌وکار تقسیم خواهند شد. این سازمان‌دهی به صورت سلسله‌مراتبی است.

در چنین شرکت‌هایی به طور معمول، مسوول هر خط کسب‌وکار (که جزئی‌ترین بخش ساختار سازمانی است) اولویت‌های خود را در توسعه‌ی خط تولید نرم‌افزار در نظر می‌گیرد. اما در این شرایط، وجود یک

<sup>58</sup> orthogonal

<sup>59</sup> product division

نقشه‌ی راه برای هماهنگ کردن فعالیت‌های خطوط و واحدهای مختلف بسیار ضروری است، چرا که مثلاً اگر یکی از خطوط، نرم‌افزار مربوط به خود را تغییر دهد و به این ترتیب زمان عرضه‌ی آن را به تأخیر اندازد، به کار خطوط دیگری که ممکن است به آن نرم‌افزار نیاز داشته باشند لطمه وارد می‌شود. البته تا حد واحدهای کسب‌وکار، در نظر گرفتن یک معماری مرجع و تعبیه‌ی نقاط تغییر در آن، کفایت خواهد کرد.

مشکل عمدتاً در سطح بخش‌های مسوول محصولات به وجود خواهد آمد. در این سطح مدیران و مسوولان بخش‌ها بیش از سطوح پایین‌تر توجه خود را تنها به بخش خود معطوف می‌کنند، به جای آن که اشتراک‌ها را در نظر بگیرند. در این شرایط هماهنگ کردن فعالیت‌های بخش‌ها و انطباق آن‌ها بر نقشه‌ی راه بسیار دشوار خواهد بود. به بیان دیگر این بخش‌ها عملاً کمابیش به صورت شرکت‌های مستقلی درمی‌آیند و با وجودی که به اشتراک‌گذاری نرم‌افزار بین آن‌ها چندان به صرفه نیست، اما در موقعیت‌هایی ممکن است کاملاً لازم باشد.

نویسندگان [1] سپس نتیجه گرفته‌اند که بهتر است این راه حل دنبال شود که هر یک از خطوط کسب‌وکار در شرکت، مؤلفه‌هایی را تولید کنند که در آن متخصص‌اند. آن‌گاه طرقتی یافت شود که با استفاده از آن‌ها بتوان این مؤلفه‌ها را برای حصول محصول مورد نیاز با هم «ترکیب» کرد. اما آیا باید راه‌های بالا به پایین را در تولید خانواده‌های محصولات نرم‌افزاری به تمامی کنار گذاشت و تنها به دفاع از روش‌های پایین به بالا پرداخت؟ پاسخ منفی است. ما هنوز به راه حلی نیاز داریم که تعادلی از رویکرد بالا به پایین و پایین به بالا را در کنار هم داشته باشد، چرا که می‌خواهیم مطمئن باشیم که مؤلفه‌ها به خوبی به یک‌دیگر متصل و یک‌پارچه می‌شوند.

#### ۷-۴- لزوم حفظ تعادل بین دو رویکرد بالا به پایین و پایین به بالا

چنان که می‌دانیم رویکرد سنتی به خط تولید، رویکردی معماری‌محور، بالا به پایین، برنامه‌ریزی شده، و درون‌سازمانی است. این در حالی است که رویکرد مبتنی بر ترکیب مؤلفه‌ها، مقرر می‌کند که تولید و توسعه باید مؤلفه‌محور، پایین به بالا، معطوف به فرصت‌ها<sup>60</sup>، و بین‌سازمانی باشد. جدول زیر به صورت خلاصه اطلاعاتی در این زمینه به دست می‌دهد:

بین‌سازمانی	آماده‌ی استفاده	معطوف فرصت	به	پایین به بالا	مؤلفه‌محور
معماری‌های	طراحی	با	برنامه‌سازی	اکستریم	معماری‌محور

<sup>60</sup> opportunistic

دامنه	استفاده‌ی مجدد			بالابه‌پایین
قرارداد واگذاری <sup>۶۱</sup>		تعیین نقشه‌ی راه		برنامه‌ریزی شده
		طراحی برای استفاده‌ی مجدد		در دست تولید
	جمعیت محصولات			درون‌سازمانی

چنان که در جدول بالا دیده می‌شود، با وجودی که رویکردهای متضاد در دو بعد به چشم می‌خورد، اما همه‌ی آن‌ها به هم وابسته نیستند و بنابراین می‌توان ترکیبی از آن‌ها را اتخاذ کرد. در جدول بالا:

- «برنامه‌سازی اکستریم» روشی معطوف به فرصت‌ها و پایین‌به‌بالا، ولی در عین حال معماری محور است.
- توسعه‌ی بالابه‌پایین و معماری محور نرم‌افزارها گاهی «طراحی با استفاده‌ی مجدد» خوانده می‌شود، در حالی که از توسعه‌ی پایین‌به‌بالا و مؤلفه‌محور اغلب «طراحی برای استفاده‌ی مجدد» نام می‌برند.
- رویکرد بالابه‌پایین و معماری محور تنها در صورتی می‌تواند سازمان‌های مختلف را دربرگیرد که اجماعی بر سر مفهوم «معماری دامنه» وجود داشته باشد.
- توسعه‌ی برنامه‌ریزی‌شده‌ی نرم‌افزارها که به شکل بین‌سازمانی انجام شود، اغلب «قرارداد واگذاری» نامیده می‌شود، و نباید با توسعه‌ی نرم‌افزارهای قابل استفاده‌ی مجدد اشتباه گرفته شود.

در [1] نویسندگان راه حلی را مناسب می‌دانند که مؤلفه‌محور، پایین‌به‌بالا، و معطوف به فرصت باشد، و در عین حال به شکل حداکثری از دارایی‌های نرم‌افزاری موجود استفاده کند تا از این طریق بتوان محصولات را «داخل سازمان» تولید کرد. این رویکرد را مبتنی بر «جمعیت محصولات» می‌دانند. به بیان دقیق‌تر ضمن این که رویکرد مؤلفه‌محور و پایین‌به‌بالاست، از معماری سبک‌وزنی نیز استفاده می‌شود؛ چرا که در غیر این صورت اشتباهات معماری می‌تواند مشکل‌آفرین باشد. به علاوه، ضمن دید معطوف به فرصت در استفاده از دارایی‌های موجود، در تولید نرم‌افزارهای جدید برنامه‌ریزی‌شده و مبتنی بر یک نقشه‌ی راه باید عمل کرد. در نهایت رویکرد مورد نظر باید درون‌سازمانی باشد، و در عین حال واحدهای زیادی را درگیر کند.

<sup>61</sup> subcontracting

سرانجام باید به این سوال پاسخ داده شود که در این راه حل استفاده از محصولات دیگران چه جایگاهی دارد؟ پاسخ این است که استفاده از محصولات دیگران انجام می‌شود، اما قطعاً به آن‌ها محدود نخواهد بود. نخست آن که محصولات دیگران اغلب عام‌منظوره است و ویژه‌ی دامنه‌ای خاص تهیه نشده، بنابراین هم‌چنان باید بخش بزرگی از مؤلفه‌های مورد نیاز را خودمان تهیه کنیم. دوم این که باید بین انواع مختلف مؤلفه‌های نرم‌افزاری تفاوت قایل بود. طبیعی است که نرم‌افزارهای پایه‌ای را باید از دیگران بگیریم، منتهمی مؤلفه‌های نرم‌افزاری مرکزی<sup>۶۲</sup> را مشخصاً باید خودمان بسازیم؛ چرا که قاعدتاً معدود شرکت‌هایی تخصص لازم برای تولید آن‌ها را دارند. البته این مؤلفه‌ها می‌توانند داخل شرکت به اشتراک گذاشته شوند. سوم این که اغلب مؤلفه‌های محصول دیگران، مؤلفه‌هایی کوچک‌اند؛ در حالی که نیاز اصلی به مؤلفه‌های نسبتاً بزرگ‌تر است. در نهایت باید گفت که برای پذیرفته‌شدن در بازار محصولات، نیاز است که پیاده‌سازی عمل‌کردهای مورد نیاز با توجه به قواعد و اجماع دیگران صورت گیرد.

#### ۷-۵- ترکیب دقیقاً به چه مناسبت؟

ترکیب را چنین تعریف می‌کنیم که دو قطعه نرم‌افزار (مؤلفه) که هر کدام بدون هیچ دانشی از دیگری تولید شده‌اند، بتوانند به یک‌دیگر متصل شوند تا یک محصول قابل استفاده را بسازند [1]. برای این منظور، این مؤلفه‌ها ممکن است در یک بستر و زمینه‌ی مشترک تولید شوند، یا واسطی را بین خود تعریف کنند. واضح است که در هر صورت دو مؤلفه باید حداقلی از اشتراک داشته باشند تا بتوانند به صورت کارا ترکیب شوند و همکاری کنند. در هر حال، نکته‌ی مهم این است که مؤلفه‌ی اول بدون مؤلفه‌ی دوم قابل استفاده باشد، و برعکس. مزیت اصلی رویکرد مبتنی بر ترکیب آن است که به واسطه‌ی استقلال مؤلفه‌ها، تنوع زیادی از محصولات را می‌توان با ترکیب‌های مختلف آن‌ها به دست آورد. البته باید توجه داشت که پلاگین‌ها در این تعریف نمی‌گنجد؛ چرا که کاملاً به چارچوبی که در آن استفاده می‌شوند وابسته‌اند و نمی‌توان آن‌ها را به صورت مستقل (از آن چارچوب) به کار برد.

#### ۷-۶- جمع‌بندی و مقایسه

مدیریت پیچیدگی و تنوع محصولات هدف، و در عین حال پای‌بندی به کیفیت بالا و زمان عرضه‌ی کوتاه، یکی از چالش‌های مهم صنعت توسعه‌ی نرم‌افزار است. در این راه، «استفاده‌ی مجدد»، «معماری»، و «مؤلفه‌ها» اجزای اصلی راه‌حلی هستند که اغلب اتخاذ می‌شوند. خطوط تولید بر همین اساس به وجود آمدند، اما در ابتدا گوناگونی‌ها به صورت محدود پوشش داده می‌شد، و تمرکز عمدتاً بر تعریف یک معماری فارغ از گوناگونی‌ها (مبتنی بر اشتراک‌ها) بود. با این حال به تدریج با گسترش محدوده‌ی محصولات، و

<sup>62</sup> core



فراتر رفتن تولید و توسعه از مرزهای سازمان، راه حل‌های دیگری پیش‌نهاد شده است که بتواند تنوع و پیچیدگی فزاینده‌ی این محصولات را مدیریت کند. رویکرد مبتنی بر ترکیب، یکی از این پیش‌نهادهاست.

البته این به معنی بازگشت به دوران تکیه بر محصولات دیگران نیست، بلکه تعادلی بین رویکرد بالابنه‌پایین و معماری محور، و رویکرد پایین‌به‌بالا و مؤلفه‌محور لازم خواهد بود. علاوه بر این، ترکیب به تمامی جای‌گزین تغییرپذیری نخواهد شد، بلکه باید این‌ها را دو بعد متفاوت در نظر آورد. در [۱] گفته شده است که در شرکت فیلیپس، حرکت به سمت رویکرد مبتنی بر ترکیب نتایج خوبی داشته است و تنوع، بهتر مدیریت شده، و هم‌چنین گستره‌ی وسیع‌تری از محصولات تولید شده‌اند. در عین حال بررسی نویسندگان منبع یادشده نشان می‌دهد که هنوز شرکت‌های کمی برای حرکت به سمت این رویکرد تمایل پیدا کرده‌اند، ولی همان‌ها به تولید گروه بزرگی از محصولات، و فراتر رفتن از مرزهای سازمان‌های داخلی دست زده‌اند و در این مسیر به موفقیت رسیده‌اند.

به این ترتیب با توجه به آنچه تا این جا گفتیم، به نظر می‌رسد که یکی از شرکت‌هایی که این رویکرد را در ایران اتخاذ کرده، شرکت اعوان است. سیاست‌های و اهداف این شرکت، به میزان زیادی با شرایطی که در پژوهش بررسی‌شده در بالا توصیف شده، تطابق و مشابهت دارد. در ادامه به این موارد خواهیم پرداخت.

نخست باید گفت که در هر دو مورد، هدف‌گذاری برای تولید گروه بزرگی از محصولات، شرکت‌ها را به کاربرد این روش متمایل کرده است. به تعبیری توسعه‌ی محدوده از حدود سنتی خطوط تولید انگیزه بوده است. همان‌طور که گفته شد شرکت اعوان عرضه‌ی خط تولید به کمک خط تولید را در دستور کار خود قرار داده، و به این ترتیب به نوعی به سمت جمعیت محصولات گرایش پیدا کرده است. در هر دو مورد، تجربه‌های عملی و واقعیت‌های بازار موجب این گرایش شده است.

دیگر این که در هر دو مورد، معماری مرجع یا چیزی مشابه آن، محور کار قرار نگرفته است، بلکه چنان که دیدیم معماری عمدتاً به جهت سامان‌دادن به تولید استفاده شده است تا مؤلفه‌ها با نقشه‌ای مشخص در کنار هم قرار گیرند و محصول مورد انتظار را محقق کنند. در واقع، در این رویکرد از معماری مرجع خبری نیست؛ بلکه معماری سبک‌وزنی متناسب با محصولات تعریف خواهد شد. این معماری مؤلفه‌های مورد نیاز و آرایش آن‌ها را مشخص خواهد کرد.

تغییرپذیری در هیچ‌کدام از این دو نفی نشده بلکه در جهت مناسب‌سازی مؤلفه‌ها به کار گرفته شده است. به علاوه، با توجه به توضیحات شرکت اعوان، دارایی‌های نرم‌افزاری مختلف، توسط گروه‌های متخصص تولید هر کدام ساخته می‌شوند؛ همان‌طور که در مورد پژوهش بررسی‌شده، بر چنین موضوعی تأکید شده است.

استفاده از محصولات دیگران، موضوع دیگری است که به نظر می‌رسد در هر دو مورد به طور محدود و در سطح نرم‌افزارهای پایه‌ای (نظیر سیستم عامل‌ها) نه تنها مجاز دانسته می‌شود، بلکه آن را ضروری تشخیص می‌دهند. با وجود این محصولات و دارایی‌های نرم‌افزاری تخصصی‌تر در هر دو مورد توسط خود شرکت ساخته می‌شود. در واقع باید گفت که ضرورت استفاده‌ی مجدد از محصولات دیگران برای هر دو محرز است؛ با این حال، اگر از دیدگاه FEF به ماجرا نگاه کنیم، هیچ‌کدام توقف در سطح ۲ جنبه‌ی «استفاده‌ی مجدد» را مطلوب نمی‌دانند.

بنابراین با عنایت به آنچه گفته شد می‌توان چنین نتیجه گرفت که رویکرد اتخاذشده توسط شرکت اعوان، رویکردی به مراتب جدید است و نمی‌توان صرف این که در قالب تعاریف و معیارهای ارزیابی سنتی خطوط تولید نمی‌گنجد، حکم به ضعیف‌بودن خط تولید این شرکت از بعد معماری داد. در واقع چنین به نظر می‌رسد که سیاست‌گذاران شرکت اعوان، با استفاده از تجربیات خود و رصد بازار به همان نتیجه‌ای رسیده‌اند که شرکت بزرگی مثل فیلیپس نیز در جهت مدیریت پیچیدگی و تنوع محصولات، و در یک کلام گسترش محدوده‌ی محصولات هدف خود، به آن رسیده است.

بحث در این باب را با شرح مختصر کاربرد رویکرد مبتنی بر ترکیب در یک نمونه‌ی دیگر به پایان می‌بریم.

## ۷-۷- یک مثال: SPLI

زیرساخت SPLI یا زیرساخت خط تولید نرم‌افزار<sup>۶۳</sup> به جهت افزایش استفاده‌ی مجدد و تغییرپذیری در یک جمعیت محصولات ارائه شده است [2]. در این زیرساخت، یک رویکرد پایین‌به‌بالا که در آن ویژگی‌های محصولات در قالب تعدادی مؤلفه با قابلیت استفاده‌ی مجدد زیاد محقق می‌شود. این مؤلفه‌ها «مؤلفه‌های فعال<sup>۶۴</sup>» نامیده شده‌اند. علت این نام‌گذاری آن است که به شکل فعال در فرآیند تولید محصول نهایی شرکت می‌کنند. علاوه بر این با اتخاذ رویکرد مهندسی مدل‌محور، تغییرپذیری و هم‌چنین طراحی مربوط به برنامه‌های کاربردی به کمک مدل‌های مربوط به دامنه، و مدل‌های تغییرپذیری توصیف می‌شوند.

در معرفی SPLI دو فرض در نظر گرفته شده است: نخست آن که هیچ محصول نرم‌افزاری مرکزی یا محوری یا به شکل میانجی در جمعیت محصولات وجود ندارد، و دوم آن که ویژگی‌ها در تعداد زیادی از محصولات نامتجانس به شکل مجدد استفاده می‌شوند. علت این امر، مشابه پژوهشی که پیش‌تر بررسی کردیم، غلبه بر چالش پیچیدگی، در نتیجه‌ی افزایش زیاد تعداد ویژگی‌هاست.

به این ترتیب بنا بر [2]، به کمک SPLI سه هدف زیر محقق می‌شود:

<sup>63</sup> Software Product Line Infrastructure

<sup>64</sup> active components

• کاهش پیچیدگی، با معرفی مؤلفه‌های فعال و سازمان‌دهی ویژگی‌ها در قالب این مؤلفه‌ها، که می‌توانند به شکل سلسله‌مراتبی ترکیب شوند.

• افزایش تغییرپذیری، به کمک مدل‌های مربوط به دامنه و سازوکارهای مولد برای تغییرپذیری

• محقق‌سازی گام‌به‌گام<sup>۶۵</sup> طراحی برنامه‌های کاربردی، از طریق کاربرد ترکیب، و ویژه‌سازی<sup>۶۶</sup>

برای رسیدن به این هدف‌ها، رویکرد کلی همان است که پیش‌تر هم گفتیم؛ یعنی حرکت از تغییرپذیری محض، به سمت مخلوطی از ترکیب و تغییرپذیری. به گفته‌ی [2] با انتقال ویژگی‌ها (که عامل پیچیدگی هستند) از معماری به مؤلفه‌های مستقل، از پیچیدگی بیش از معماری جلوگیری کرده‌ایم. به این ترتیب با ره‌یافتی مؤلفه‌محور و پایین‌به‌بالا، تنها انتخاب مؤلفه‌هاست که ویژگی‌های یک محصول نرم‌افزاری را محقق خواهد کرد.

ذکر این نکته لازم است که با توجه به [2] این ویژگی‌ها می‌توانند انواع گوناگونی از مصنوعات، از قبیل کد منبع، مدل‌ها، فایل‌های پیکربندی، فایل‌های داده‌ای و تصاویر را دربرگیرند. به این ترتیب مؤلفه‌هایی که یک ویژگی را محقق می‌کنند می‌بایست شامل انواع گوناگونی از مصنوعات نرم‌افزاری در سطوح انتزاعی متفاوتی باشند. به علاوه، انتشار سازگار تغییرپذیری بین مصنوعات مختلف، وظیفه‌ای است که بر دوش مؤلفه‌نهاد شده است، و به همین خاطر است که چنان که بعداً خواهیم دید، «تبدیل»<sup>۶۷</sup> بر عهده‌ی خود مؤلفه‌ها گذاشته شده تا از یک پارچگی داخلی در نتیجه‌ی استفاده‌ی مجدد اطمینان حاصل کنند.

بنابراین طراحی برنامه‌های کاربردی در بستر SPLI را می‌توان بیان تغییرپذیری در مدل‌های مربوط به دامنه دانست. به بیان دیگر برنامه‌ی کاربردی، خود مؤلفه‌ای است که قرار است توقعات از محصول نهایی را برآورده کند. این جاست که مرز بین برنامه‌ی کاربردی و مؤلفه قدری باریک می‌شود.

مؤلفه‌های فعال (چنان که پیش‌تر گفته شد) فعالانه در فرآیند تولید محصول نهایی مشارکت می‌کنند. این مشارکت از طریق فراهم‌کردن واسطی است که از طریق آن می‌توان به مؤلفه فرمان داد که خروجی مورد نیاز را برای فرآیند اشتقاق آماده کند. به این ترتیب همه‌ی مؤلفه‌ها می‌بایست این واسط را فراهم کنند و به این ترتیب نوعی چندشکلی<sup>۶۸</sup> در این جا برقرار است. علت این امر عمدتاً به همان فرضی مربوط می‌شود که پیش‌تر گفته شد؛ یعنی این که ویژگی‌ها در محصولات نامتجانسی وجود دارند و لذا استقلال و خودکفایی

<sup>65</sup> step-wise refinement

<sup>66</sup> specialization

<sup>67</sup> transformation

<sup>68</sup> polymorphism

آن‌ها از منتهای اهمیت برخوردار است. از سوی دیگر از آن‌جا که در این رویکرد عنصر محوری وجود ندارد، وظیفه‌ی بررسی یک‌پارچگی و صحت کارکرد، از معماری به خود مؤلفه‌ها واگذار شده است.

مدل‌های مبتنی بر مهندسی مدل‌محور ابزار دیگری است که برای رسیدن به اهداف سه‌گانه‌ی ذکرشده در بالا به کمک ما می‌آید. این مدل‌ها در جریان اشتقاق محصول نهایی، به سادگی قابل تبدیل به مصنوعات سطح پایین‌تر (مثلاً فایل‌های کد منبع) هستند. تسریع تولید و توسعه، کیفیت نرم‌افزار، جداسازی مسوولیت‌ها<sup>69</sup>، استفاده‌ی مجدد، و مدیریت پیچیدگی به کمک انتزاع، از جمله مهم‌ترین مزایای روش مدل‌محورند، که اتفاقاً بر انگیزه‌های حرکت به سمت خط تولید نیز منطبق‌اند. در رویکرد مطرح‌شده در [2] مدل‌های مربوط به دامنه، به جای مدل‌های فراگیر (مثل UML) مد نظر قرار گرفته‌اند.

به طور خلاصه باید گفت که SPLI به صورت ترکیبی از مؤلفه‌ها (ویژگی‌ها)، و تغییرپذیری‌هایی که در مدل‌های تغییرپذیری یا مدل‌های مربوط به دامنه توصیف شده‌اند، قابل تعریف است، و مراحل اشتقاق یک محصول در آن به قرار زیر است [2]:

۱. **گردآوری همه‌ی مؤلفه‌های فعال مورد نیاز:** اشاره‌گری به همه‌ی مؤلفه‌های موجود در ترکیب‌بندی، در آن تعریف شده است. از این طریق همه‌ی مؤلفه‌های فعال مشخص‌شده در آن، گردآوری می‌شوند تا از وجود آن‌ها برای ادامه‌ی مراحل اطمینان حاصل شود.

۲. **انتخاب، ترکیب، و ویژه‌سازی مدل‌ها:** در این مرحله همه‌ی مدل‌ها در داخل مدل یکتای برنامه‌ی کاربردی ادغام و تجمیع می‌شوند. در این ادغام، ویژه‌سازی مدل‌ها نیز لحاظ شده است. این مدل‌ها می‌توانند از انواع گوناگونی باشند، و تنها مدل یکتای برنامه، بنابر SPLI، می‌بایست مبنای تمام مراحل (و تبدیلات) بعدی قرار گیرد. به علاوه این ادغام در یک مدل، امکان مواجهه با مجموعه‌های احتمالاً نامتجانس نرم‌افزارها را فراهم می‌کند.

۳. **اجرای تبدیلات مدل به مدل:** در این مرحله، مدل یکتای مرحله‌ی پیشین توسط مؤلفه‌های فعال دریافت، و تحلیل، می‌شود و تغییرات لازم به اجرا در می‌آید. در واقع این مرحله فرصت آخر مؤلفه‌ها برای انجام تغییرات در راستای برآورده کردن نیازمندی‌های مورد انتظار از مؤلفه نیز هست.

۴. **انجام تحلیل ایستا<sup>70</sup>:** در این مرحله، تحلیلی ایستا (در مقابل تحلیل زمان اجرا) توسط مؤلفه‌ها انجام می‌شود تا این اطمینان به وجود آید که مدل از نظر معنایی صحیح است، و نیازمندی‌ها برآورده

<sup>69</sup> separation of concerns

<sup>70</sup> static analysis

شده‌اند. روشن است که اگر در این تحلیل خطایی مشخص شود، فرآیند به کلی متوقف می‌شود و موضوع به اطلاع کاربر خواهد رسید.

۵. **اجرای تبدیلات مدل به مصنوع:** در این مرحله، کنترل فرآیند در اختیار مؤلفه‌های فعال قرار می‌گیرد تا با دریافت مدل یکتای برنامه‌ی کاربردی به عنوان ورودی، مصنوعات مورد انتظار را به عنوان خروجی تحویل دهند.

۶. **انتخاب مصنوعات نرم‌افزاری ایستا:** در این مرحله، از مؤلفه‌های فعال خواسته می‌شود که مصنوعات نرم‌افزاری ایستا (در مقابل تغییرپذیر یا تولیدشده توسط مؤلفه‌ها. مثلاً کتاب‌خانه‌ها یا قطعه‌های کد از آن جمله‌اند) را انتخاب کنند تا با مصنوعات دیگر یک‌پارچه شوند. مؤلفه‌ها می‌توانند از مدل برنامه‌ی کاربردی، و به ویژه مدل‌های تغییرپذیری برای انتخاب دقیق‌تر ویژگی‌ها و مؤلفه‌های لازم بهره‌جویند.

۷. **تجمیع مصنوعات نرم‌افزاری در برنامه‌ی کاربردی:** در این مرحله، همه‌ی مصنوعات نرم‌افزاری انتخاب‌شده (اعم از تولیدشده به وسیله‌ی مؤلفه‌ها، یا مصنوعات ایستا) در برنامه‌ی کاربردی نهایی تجمیع می‌شوند. فرآیندی که معمولاً شامل کپی کردن فیزیکی فایل‌های لازم در فضای مشخص شده برای برنامه‌ی کاربردی است.

۸. **کامپایل:** مرحله‌ی آخر کامپایل همه‌ی کدها (اعم از مصنوعات ایستا یا تولیدشده) خواهد بود.

آنچه در این فرآیند مهم است، واگذاری قدرت تصمیم‌گیری بی‌اندازه به مؤلفه‌هاست. در واقع فرآیند اشتقاق تنها رویه‌های مربوط را در مؤلفه‌ها فرامی‌خواند و حاصل کار را در جایی ذخیره می‌کند. به این ترتیب این مؤلفه‌ها هستند که باید با توجه به مدل برنامه‌ی کاربردی انتخاب‌ها و تغییرات را به درستی انجام دهند.

طراحان در ارزیابی خود از SPLI گفته‌اند که ترکیب در این زیرساخت، نوعی انتخاب ویژگی است، و بنابراین امکان تغییرپذیری را می‌دهد [2]. به علاوه مشکل را در رویکرد معماری محور خطوط تولید در این دانسته‌اند که همه‌ی تغییرپذیری‌های مورد نیاز باید تنها در «یک» معماری مرجع پیش‌بینی شود و قابل پیاده‌سازی باشد. البته در عین حال این نکته را نیز متذکر شده‌اند که فقدان مطلق معماری در زیرساخت معرفی شده، یکی از نقاط ضعف آن است؛ چرا که مؤلفه‌ها تنها به شکل ضمنی نوعی معماری را در نظر می‌گیرند، اما این معماری نمی‌تواند مشکلات احتمالی را بررسی کند. هرچند مشکلات آشکار معماری و ساختاری در مرحله‌ی نهایی تجمیع و کامپایل مشخص می‌شوند، بسیاری از مشکلات نیز ممکن است پنهان بمانند. از سوی دیگر، تبدیل‌ها نیز می‌توانند مانع ردگیری تغییرپذیری‌ها شوند که این خود کاستی دیگری است.

## ۸- توصیه‌ها به شرکت اعوان

### ۸-۱- مقدمه

این فصل در واقع خلاصه‌ای است از نتایج ارزیابی‌هایی که در قالب FEF صورت گرفت. هدف از ارائه این فصل، راهنمایی‌هایی است که به شرکت اعوان کمک می‌کند تا از سطح بلوغ فعلی به سطح بعدی برسد.

### ۸-۲- ربع کسب و کار

- نتایج ارزیابی FEF بر روی اعوان نشان داد که نتایج و منافع بدست آمده از خط تولید به صورت کمی مدل‌سازی نشده است. بنابراین لازم است که اعوان در رویکردهای راهبردی خود بخش جدی‌تری را صرف مدل‌سازی کمی از میزان تاثیر تصمیم‌گیری‌های مختلف بر روی خط تولید خود کند تا به نتایج دلخواه برسد.

### ۸-۳- ربع معماری

- اعوان نمره ضعیفی در زمینه معماری خط تولید کسب کرده بود. در نتیجه باید اهمیت بیشتری به طراحی مدل مرجع خود که شالوده معماری خط تولید را می‌سازد بدهد و نقاط تغییر و متغیرها را به صورت جدی‌تر در این مدل مرجع وارد کند.

### ۸-۴- ربع فرآیند

- فرآیندهای اعوان تا حد خوبی از بلوغ رسیده بودند. با این وجود نیاز به کمی‌سازی بهتر و اندازه‌گیری میزان کارایی آن به منظور مطابقت با مدل CMMI احساس می‌شود. با این وجود اعوان نشان داده که برای بهینه‌سازی فرآیندها در جهت اهداف خط تولید حاضر به سرمایه‌گذاری است.

### ۸-۵- ربع سازمان

- با توجه به کوچک بودن تیم نرم‌افزاری اعوان، میزان به اشتراک‌گذاری دانش بخش core و برنامه‌ها در حد قابل قبولی بود. با این وجود با گسترده‌تر شدن اندازه تیم نیاز جدی‌تری به وجود نقش‌هایی مانند wise-mentor در تمامی پروژه‌های در دست اجرای اعوان مشاهده می‌شود.

- [1] R. V. Ommerring, J. Bosch, "Widening the Scope of Software Product Lines –From Variation to Composition", Proceedings of the Second Software Product Line Conference, Springer Verlag, Berlin Heidelberg, pp. 328-347, 2002
- [2] B. Geertsema, S. Jansen, "Increasing Software Product Reusability and Variability using Active Components: a Software Product Line Infrastructure", Proceedings of the Fourth European Conference on Software Architecture, ACM, New York, USA, pp. 336-343, 2010
- [3] F. van der Linden, "Family Evaluation Framework, overview and introduction", Philips Medical Systems, version, 2005
- [4] F. van der Linden, K. Schmid, E. Rommes, Software Product Lines in Action, Springer, 2005
- [5] <http://asta.ir/avan/>