

## مدیریت تیم توسعه‌ی نرم‌افزارهای ستادی

- (۱) حسین شریفی‌پناه  
(۲) مهدی مشایخی  
(۳) فریبرز کمالی تبریزی  
(۴) علیرضا امیدوار

<sup>۱</sup> مسئول فنی تیم توسعه‌ی نرم‌افزار، پژوهشگاه صنعت نفت  
<sup>۲</sup> مسئول تیم توسعه‌ی نرم‌افزار، پژوهشگاه صنعت نفت  
<sup>۳</sup> رئیس فناوری اطلاعات و مدیریت دانش، پژوهشگاه صنعت نفت  
<sup>۴</sup> رئیس شبکه‌های رایانه‌ای و خدمات مکانیزه، پژوهشگاه صنعت نفت

<sup>1</sup> sharifipanah@ripi.ir

<sup>2</sup> mashayekhi@ripi.ir

<sup>3</sup> kamali@ripi.ir

<sup>4</sup> omidvara@ripi.ir

واژه‌های کلیدی: مدیریت تیم توسعه‌ی نرم‌افزار، متدولوژی توسعه‌ی نرم‌افزار

### چکیده

جایگزینی یک متدولوژی جدید با رویکردهای قدیمی در محیط توسعه نرم‌افزار همیشه با چالش‌های فراوانی همراه بوده است. یکی از مهم‌ترین سؤوال‌هایی که قبل از شروع این فرایند پرسیده می‌شود این است که آیا این متدولوژی مناسب با تیم توسعه نرم‌افزار است؟ در این تجربه مدیریتی تمامی فرایندهای تصمیم‌گیری، چگونگی انتخاب راه حل، مراحل اجرا به همراه توالی و ترتیب هر مرحله بیان گردیده است. شواهد و نتایج نشان می‌دهد که راه‌حل انتخابی به همراه مراحل اجرا در جهت تقلیل مشکلات تیم توسعه نرم‌افزار اثربخش بوده‌اند.

## ۱- مقدمه

ظهور متدولوژی‌های گوناگون در عرصه تولید نرم‌افزار از این جهت بوده است که نیاز تیم‌های توسعه به همراه سطح توقع متقاضیان در گذر زمان در حال تغییر بوده است. لذا یک متدولوژی جای خود را به متدولوژی دیگری می‌دهد. جایگزینی یک متدولوژی با متدولوژی دیگر پایان مجادله‌ها نیست بلکه شروع چالش‌ها است.

هر متدولوژی علاوه بر این که مجموعه‌ای از توصیه‌ها برای تغییر رویکردها به همراه دارد، سعی می‌کند فرهنگ حاکم بر تیم را نیز تحت‌الشعاع قرار دهد. عمل به توصیه‌ها مستلزم حاکم شدن فرهنگ متدولوژی جدید است. نکته مهمی که در حاکم شدن فرهنگ متدولوژی وجود دارد این است که این فرهنگ باید با در نظر گرفتن شرایط موجود در تیم توسعه نرم‌افزار جایگزین فرهنگ قبلی شود. بنابراین تمامی موارد مطرح شده در توصیه‌های یک متدولوژی لزوماً قابل اجرا نخواهند بود.

آنچه در این تجربه مدیریتی مورد بررسی قرار گرفته است، ریشه‌یابی مشکلات موجود در تیم توسعه نرم‌افزار می‌باشد که منجر به انتخاب یک متدولوژی جدید به عنوان راه‌کار مرتفع نمودن مشکلات گردیده است. همانطور که بیان شد جایگزینی این متدولوژی جدید با شناخت محیط تیم توسعه و آشنایی با ریسک‌ها و تهدیدها می‌توانست کارساز باشد. لذا آنچه در این مقاله بیان می‌شود نحوه‌ی تصمیم‌گیری در این زمینه است.

در این مقاله پس از بیان پیشینه‌ای از سازمان مورد بحث و ساختار سازمانی آن به بیان مسأله و ضرورت خواهیم پرداخت. پس از آن تصمیم‌گیری و چگونگی انتخاب راه‌حل بیان خواهد شد. مراحل اجرا به همراه توالی و ترتیب هر مرحله مورد اشاره قرار خواهد گرفت. در نهایت با ارائه شواهد نشان خواهیم داد که تصمیمات اتخاذ شده اثربخش بوده‌اند.

## ۲- زمینه و موقعیت تجربه

پژوهشگاه صنعت نفتی یک از بزرگ‌ترین مراکز تحقیقاتی در منطقه خاورمیانه می‌باشد که فعالیت‌های این مرکز با توجه به اهمیت نفت و انرژی‌های نوین در اقتصاد ایران، از حساسیت بالایی برخوردار است. بنابراین فراهم آوردن ابزارهایی در جهت تسهیل کسب و کار این مرکز تحقیقاتی در اولویت بالایی قرار دارد. یکی از ابزارهایی که در این زمینه می‌تواند نقش بسیار مهمی ایفا کند، سیستم‌های نرم‌افزاری Enterprise می‌باشند که مجموعه شبکه‌های رایانه‌ای و خدمات مکانیزه در پژوهشگاه صنعت نفت متولی این امر می‌باشد.

از آنجاییکه شروع توسعه‌ی یک بخش نرم‌افزاری با اعلام نیاز از سوی مجموعه‌های مختلف پژوهشگاه صورت می‌پذیرد و در بازه‌های زمانی مختلف تعداد تقاضاها مختلف است، لذا مجموعه توسعه دهنده نرم‌افزار دارای ساختار متفاوت نیروی انسانی مستقر در بازه‌های زمانی مختلف است. این تفاوت ساختار در بازه‌های زمانی مختلف هم از نظر تعداد افراد و هم از نظر نقش آنها اتفاق می‌افتد. نیاز به داشتن وحدت رویه در پیشبرد اهداف سازمان دلیلی است که منجر به استقرار مسئول ثابت در این مجموعه توسعه دهنده نرم‌افزار گردیده است. شروع توسعه یک بخش نرم‌افزاری جدید، پس از آنالیز و بررسی توجیه تولید آن توسط تیم توسعه دهنده داخلی پژوهشگاه صنعت نفت، مستلزم وجود منابع نیروی انسانی کافی برای آن می‌باشد. لذا در شرایطی که شروع توسعه یک بخش نرم‌افزاری جدید در اولویت است و منابع نیروی انسانی کافی نمی‌باشند، این مجموعه با استخدام افراد به صورت قراردادی اقدام به پیشبرد اهداف خود می‌نماید.

## ۳- مساله و ضرورت

طبق انتظار شروع توسعه یک بخش نرم‌افزاری جدید با ارائه زمان‌بندی براساس منابع انسانی موجود همراه بوده است. معمولاً هر نیروی انسانی در زیرسیستمی از نرم‌افزار همکاری می‌کردند. لذا در صورتیکه توسعه یک زیرسیستم بدلیل پیچیدگی و حجم کار نیاز به بیش از یک نیروی انسانی داشت، شخصی که بیشترین تجربه را داشت به عنوان سرپرست کار انتخاب می‌شد. کنترل زمان‌بندی‌ها معمولاً برعهده این شخص بود. جدا از اینکه بدلیل شفاف نبودن وضعیت منابع در اختیار، زمان‌بندی‌ها غیرواقع‌بینانه بود، تجربه نشان داده بود که بطور متوسط زمان‌بندی وظایف بیش از ۵۰ درصد زمان تخمین اولیه تمدید می‌شد. بنابراین نارضایتی متقاضیان بدلیل تأخیر در انجام وظایف محوله اولین مشکلی بود که بر سر راه تیم توسعه نرم‌افزار وجود داشت.

از طرف دیگر همانطور که ذکر شد بدلیل اینکه هر توسعه دهنده در توسعه زیرسیستم‌های خاصی نقش ایفا می‌کرد و همچنین استاندارد مشخصی برای مستندسازی فعالیت‌ها تعریف نشده بود، احتمال اینکه ناهماهنگی در طراحی و پیاده‌سازی بوجود آید بسیار فراوان بود. این عدم هماهنگی بستری برای تولید خطاهای ناخواسته در یک سیستم Enterprise بود. علاوه بر این، این رویکرد زمینه‌ساز یکی از بزرگ‌ترین ریسک‌هایی بود که سازمان با آن مواجه بود. از آنجاییکه ترکیب اصلی تیم توسعه شامل افراد قراردادی بود، لذا انحصاری شدن زیرسیستم‌ها برای توسعه دهندگان منجر به کاهش انعطاف‌پذیری سازمان در شرایط خاص می‌گردید.

مشکلات تیم توسعه نرم‌افزار وقتی آشکارتر می‌شد که تغییری در ساختار تیم از نظر نیروها بوجود می‌آمد. نبود دانش مکتوب براساس یک استاندارد باعث دوباره کاری در انجام وظایف می‌گردید. گاهی از اوقات نداشتن شناخت درست نسبت به یک بخش از طراحی یا تکنولوژی ارائه شده منجر به تصمیمات اشتباه یا خطاهای ناخواسته در پیاده‌سازی نرم‌افزارهای جدید می‌شد.

همانطور که قبلاً نیز اشاره شد یکی از دلایل نارضایتی متقاضی تأخیر در تحویل محصولات نرم‌افزاری به متقاضی بود. بررسی بازخوردهای متقاضی در کنار عملکرد تیم توسعه نرم‌افزار حاکی از این مسأله بود که حتی در مواقعی که تأخیر در تحویل محصولات وجود نداشت یا قابل چشم‌پوشی بود، باز هم عدم رضایت متقاضی از عملکرد تیم توسعه قابل مشاهده بود. نتایج بررسی‌های صورت گرفته حاکی از آن بود که محصول تولید شده با نیاز اولیه اعلام شده از سوی متقاضی هم‌خوانی ندارد. به عبارت دیگر در رویکرد تیم توسعه نرم‌افزار پدیده تغییر مداوم نیاز متقاضی در این حوزه مورد توجه قرار نگرفته بود.

عدم شفاف بودن فرایندهای کاری اثر تمامی مشکلات مطرح شده را چندین برابر می‌کرد. زیرا در شرایط بحرانی تیم راه‌کار خاصی برای مقابله با بحران را پیش‌بینی نکرده بود. از طرف دیگر متقاضی نمی‌دانست چه نقشی در فرایند توسعه نرم‌افزار دارد. این مسأله باعث می‌شد که نقش حیاتی متقاضی در فرایند توسعه نرم‌افزار که منجر به کیفیت بالای نرم‌افزار می‌شد نادیده گرفته شود.

تمامی مشکلاتی که تاکنون مطرح شدند به صورت مستقیم بر روی کیفیت محصول خروجی تیم توسعه نرم‌افزار تأثیرگذار بودند. اما یکی از مشکلاتی که تیم توسعه نرم‌افزار با آن مواجه بود و بصورت غیرمستقیم بر روی کیفیت محصولات تأثیرگذار بود، عدم امکان ارزیابی عملکرد اعضای تیم بود.

تمامی این مشکلات در طی چند سال اخیر به قدری بر عملکرد تیم توسعه نرم‌افزار تأثیر منفی گذاشته بود که ریشه‌یابی مشکلات و انتخاب راه‌کاری مناسب برای آن‌ها لازم به نظر می‌رسید.

## ۴- تصمیم‌گیری

کاربران سیستم‌های تولید شده توسط تیم توسعه نرم‌افزار به دو دسته اصلی تقسیم می‌شوند. دسته اول خود کاربران متقاضی سیستم می‌باشند. در حالیکه دسته دوم کاربرانی را شامل می‌شود که لازم است به صورت مستقیم از سیستم‌های تولید شده استفاده کنند. تیم توسعه نرم‌افزار در بازه‌های زمانی مختلف و یا به مناسبت‌های مختلف از کاربران خود بازخوردهایی را دریافت می‌کرد. جمع‌بندی این بازخوردها وجود مشکلاتی از جمله تأخیرات غیرقابل توجیه در تحویل محصولات، عدم توجه به پدیده تغییر مداوم نیاز متقاضی در حوزه نرم‌افزار، عدم شفاف بودن جایگاه متقاضی در فرایند توسعه نرم‌افزار را تایید می‌کرد.

یکی دیگر از ابزارهای مورد استفاده جهت شناخت مشکلات، استفاده از سیستم وقایع‌نگاری بوده است. این سیستم دارای قابلیت نگهداشت خطاهایی است که در زمان اجرا رخ می‌دهد. تحلیلی که بر روی این اطلاعات انجام شد، بسیاری از مشکلات از جمله عدم وجود استاندارد تدوین شده جهت طراحی، پیاده‌سازی و مستندسازی بخش‌های مختلف نرم‌افزار را مشخص می‌نمود. در نهایت بازخوردهای دریافتی از سوی اعضای تیم توسعه نرم‌افزار مبنی بر نادیده گرفته شدن فعالیت‌های موثر آنها، منجر به ضرورتی مبنی بر وجود امکان ارزیابی عملکرد اعضای تیم گردید. تجربه در زمینه متدولوژی‌های تولید نرم‌افزار [۶] نشان می‌داد که بسیاری از مشکلات در تیم توسعه نرم‌افزار با انتخاب یک متدولوژی مناسب قابل رفع خواهند بود. اینکه چه متدولوژی مناسب با سیستم توسعه بود مستلزم مطالعه راه‌کارهای جدید بود. بنابراین مطالعه و استفاده از نظرات متخصصان صاحب‌نظر راه رسیدن به کلید مشکلات تیم بود.

## ۵- مراحل اجرا

بررسی مشکلات و مطالعه متدولوژی‌های روز دنیا در زمینه تولید نرم‌افزار به تنهایی نمی‌توانست روشنگر راه باشد. مهم‌تر از همه‌ی این مسائل انطباق رویکردهای یک متدولوژی با شرایط محیطی تیم توسعه نرم‌افزار است. بنابراین در اولین گام شناخت قابلیت‌های محیط تیم توسعه نرم‌افزار به همراه استخراج فازها و ریسک‌های موجود در هر فاز صورت پذیرفت. خطوط قرمز در اجرای هر فاز اهداف تعریف شده برای آن فاز بود. بدین معنی که هیچ بازخوردی از تیم توسعه‌ی نرم‌افزار نمی‌توانست منجر به تغییر هدف فاز شود. این در حالی است که تکنیک‌های مورد استفاده در هر فاز با توجه به بازخوردهایی که از تیم توسعه‌ی نرم‌افزار در زمان‌های مختلف دریافت می‌شد، می‌توانست اصلاح یا با تکنیک دیگری جایگزین شود.

انتخاب تکنیک‌های هر فاز و ترتیب اجرای فازها با در نظر گرفتن به سه عامل زیر تعریف شدند:

اولویت رفع مشکلات: در بعضی از موارد ترتیب اجرای فازها با توجه به اینکه آن فاز چه مشکلی از مشکلات مطرح شده در بخش ۳ را مرتفع خواهد نمود و در نظر گرفتن اولویت رفع مشکل مورد نظر، مشخص شدند.

ریسک‌ها: شناخت شرایط و تیم توسعه‌ی نرم‌افزار منجر به شناخت ریسک‌هایی شده بود که می‌توانست مانعی برای دستیابی به اهداف هر فاز گردد. لذا باید تکنیک‌هایی انتخاب می‌شد که اثرات این ریسک‌ها را تقلیل دهد. از طرف دیگر تعریف ترتیبی صحیح از اجرای هر فاز می‌توانست در به حداقل رساندن اثرات ریسک‌ها در فازهای بعدی کمک نماید.

پذیرش تیم توسعه برای اجرای فاز: برخی از تکنیک‌های مورد استفاده بدلیل ناآشنایی تیم توسعه‌ی نرم‌افزار با آن‌ها از مجموعه‌ی تکنیک‌های قابل استفاده به صورت موقت حذف شدند. برای این منظور تمام تلاش بر این بوده است که تکنیک جایگزینی برای تکنیک حذف شده ارائه شود.

در مرحله‌ی بعد پس از تعریف فازها و تکنیک‌های مورد استفاده در هر فاز باید زمان اجرای هر فاز مشخص می‌شد. نکته‌ی قابل تأملی که در این زمینه وجود دارد این است که این زمان تقریبی خواهد بود. به عبارت دیگر خاتمه‌ی اجرای هر فاز وقتی مشخص می‌شود که از دستیابی به اهداف آن فاز اطمینان حاصل شود. بنابراین این زمانبندی کاملاً وابسته به محیطی است که بستر اجرای این تجربه است. آنچه در این تجربه به عنوان زمان اجرا ثبت گردید محدوده‌ی زمانی ۹ ماه است.

در ادامه هر فاز به همراه تکنیک‌های مورد استفاده در کنار ریسک‌های شناسایی شده بیان خواهد شد.

## ۵-۱- فاز اول: آشنایی با تکنیک‌های زمان‌بندی

همانطور که در بخش ۳ نیز اشاره شد، یکی از بزرگترین مشکلات موجود در تیم توسعه‌ی نرم‌افزار زمان‌بندی‌های غیرواقع‌بینانه‌ای بود که در نهایت منجر به تأخیر در تحویل محصولات به متقاضی می‌شد. از آنجا که رفع این معضل در اولویت بالایی قرار داشت فاز نخست تصمیم‌مدیریتی به این مهم اختصاص داده شد. با توجه به شناختی که از اعضای تیم موجود بود، ریسک عدم آشنایی با نحوه‌ی تخمین زمان برای فعالیت‌ها آشکار بود. لذا تکنیک‌هایی باید استفاده می‌شد که این ریسک تقلیل پیدا می‌کرد.

بدین منظور تکنیک «مشارکت تمامی اعضای تیم در زمان‌بندی» جهت آشنایی با نحوه‌ی تخمین زمان پیشنهاد گردید. در این تکنیک تمامی اعضای تیم در جلسه‌ای مشترک در مورد زمان‌بندی انجام یک فعالیت مشارکت می‌کردند. در این نشست مشترک که بسیار زمانبر نبود، سعی می‌شد با بیان جوانب مختلف درخواست ابعاد آن شفاف شده تا بتوان زمان‌بندی دقیق‌تری ارائه نمود. پس از آن هر شخص یک زمان را ارائه می‌داد و در نهایت میانگین زمان‌بندی‌های ارائه شده زمان‌بندی نهایی خواهد بود. در بعضی از شرایط به جای استفاده از میانگین، فعالیت به شخصی منتصب می‌شد که کمترین زمان را پیشنهاد داده بود.

یکی از مواردی که به عنوان پیش‌نیاز برای زمان‌بندی دقیق در نظر گرفته شده بود وجود اطلاعاتی شفاف در مورد وضعیت منابع در اختیار است. ثبت اطلاعات فعالیت‌های در حال اجرا به همراه زمان‌بندی در نظر گرفته شده و منابع تخصیص داده شده به آن‌ها تکنیکی بود که برای شفاف نمودن این اطلاعات مورد استفاده قرار گرفت. نکته‌ای که در این زمینه مورد توجه بود استفاده از ابزار مناسب و با قابلیت استفاده‌ی آسان بود بطوریکه بتوان در کمترین زمان ممکن اطلاعات توسط تمامی اعضای تیم توسعه قابل‌بروزسانی باشند.

## ۵-۲- فاز دوم: آشنایی با تولید دانش مکتوب

یکی از معضلاتی که در تیم توسعه‌ی نرم‌افزار به چشم می‌خورد دوباره‌کاری بدلیل نبود دانش مکتوب بخصوص در مورد بخش‌های چارچوب‌های نرم‌افزار بود. این مسأله نه تنها منجر به صرف منابع تیم در جهت انجام فعالیت‌های تکراری می‌گردید بلکه در بسیاری از شرایط منجر به تولید مفاهیم یکسان ولی با رویکردهای مختلف در سیستم می‌شد. این مسأله خطاهای منطقی بسیاری را رقم می‌زد. از طرف دیگر از آنجاییکه محصولات تولید شده دارای مستندات مکتوب نبودند استفاده از آن‌ها توسط شخص توسعه‌دهنده می‌توانست منجر به تولید خطاهای ناخواسته شود. زیرا توسعه‌دهنده با ساختار و مفاهیمی که منجر به تولید محصول شده است، آشنایی نداشت و در نهایت بدلیل استفاده‌ی نادرست از آن‌ها خطاهایی را ایجاد می‌نمود. تولید این نوع خطاها جدا از غیر قابل استناد جلوه دادن نرم‌افزار در چشم مشتری، شرایط تشخیص و رفع منشأ خطا را نیز بسیار سخت‌تر می‌نمود.

با توجه به این مسائل باید شرایطی فراهم می‌گردید که بتوان با در نظر گرفتن دو ریسک عمده در این فاز به نتیجه‌ی مطلوب دست پیدا کنیم. اولین ریسک موجود عدم آشنایی تیم توسعه‌ی نرم‌افزار با ابزارهای ثبت دانش در یک محیط کار تیمی بود. از طرف دیگر ثبت دانش می‌توانست فرآیندی زمان‌بر باشد. این مسأله به تنهایی می‌توانست بر روی زمان‌بندی‌های انجام یک فعالیت تأثیرگذار باشد.

با در نظر گرفتن مشکل و ریسک‌های موجود دو راهکار مورد استفاده قرار گرفت. در ابتدا با معرفی یک لیست صحنه‌گذاری انجام کار در تیم توسعه‌ی نرم‌افزار و در این لیست قراردادن مواردی از جمله مستندسازی فعالیتی که انجام شده، اعضای تیم مجبور بودند برای هر فعالیتی که ادعای انجام آن را داشتند تمامی بندها از جمله بند «مستندسازی» را تکمیل نمایند.

از سوی دیگر برای نهادینه نمودن این فرهنگ تمامی اعضای تیم باید شرح نحوه‌ی انجام وظایف خود را هر سه ساعت در سیستم ثبت می‌نمودند. ابزاری که بدین منظور استفاده شد، ابزار شرکت مایکروسافت به نام Team Foundation Server بود. این ابزار بگونه‌ای بومی‌سازی شد که اعضای تیم به راحتی می‌توانستند شرح نحوه‌ی انجام کار خود را در آن وارد نمایند. همچنین کار با این ابزار بسیار راحت بود و زمان زیادی برای وارد نمودن اطلاعات صرف نمی‌شد.

### ۳-۵- فاز سوم: ایجاد شفافیت در فرآیندهای کاری

مشکل عدم شفاف بودن فرآیندهای کاری هم می‌توانست از داخل تیم مشکل‌ساز شود و هم از خارج آن. اعضای تیم توسعه‌ی نرم‌افزار بدلیل شفاف نبودن فرآیندها در تعاملات بین تیمی و همچنین تعاملات با متقاضی دچار مشکل می‌شدند. از طرف دیگر متقاضی نیز به دلیل شفاف نبودن انتظارات تیم توسعه در انجام تعهدات نسبت به محصول تولید شده نمی‌توانستند موفق گردند. با در نظر گرفتن این مسأله فرآیندهای «انجام فعالیت» و «پشتیبانی» با در نظر گرفتن نقش تمامی اعضای تیم توسعه‌دهنده و متقاضی با جزئیات طراحی و اجرا شد. اجرای این فرآیندها نه تنها مستلزم آگاهی دقیق اعضای تیم توسعه از مراحل هر فرآیند بود، بلکه متقاضی نیز باید از نقش خود در فرآیند مطلع می‌گردید. بدین منظور قبل از سپردن هرگونه تعهدی به متقاضی از آگاهی کامل متقاضی با فرآیندهای تیم توسعه اطمینان حاصل می‌شد.

### ۴-۵- فاز چهارم: ایجاد عامل انگیزش

تمامی تغییرات در تیم توسعه‌ی نرم‌افزار نیازمند یک ضمانت اجرایی قوی بود. نتایج مثبت حاصل از فازهای بعدی به دلیل اینکه در کوتاه مدت قابل مشاهده نبودند، به تنهایی نمی‌توانست ضمانت اجرایی برای پایبندی به قوانین جدید حاکم بر تیم توسعه‌ی نرم‌افزار باشد. لذا باید راهکاری انتخاب می‌شد تا بتواند در کوتاه مدت عامل انگیزش را برای عمل به رهنمودهای جدید حاکم شده بر تیم، فراهم آورد. بدین منظور پارامترهایی جهت ارزیابی عملکرد اعضای تیم توسعه دهنده بگونه‌ای تعریف شدند که بتوان عملکرد آن‌ها را برای رسیدن به اهداف مطرح شده برای هر فاز بررسی نمود. از آنجاییکه یکی از بزرگترین ریسک‌های موجود بر سر راه این فاز عدم تمایل اعضای تیم نسبت به ارزیابی بود، ماهانه پس از ارزیابی عملکرد اعضای تیم، مبلغی به عنوان پاداش به شخص برتر ارزیابی اختصاص داده می‌شد. یکی از نکات قابل اشاره در این فاز نحوه‌ی تعریف و وزن‌دهی پارامترهای ارزیابی عملکرد اعضای تیم است. تکنیک مورد استفاده [۲] پیشنهاد می‌نماید که تمامی اعضای تیم در فرآیند انتخاب و وزن‌دهی پارامترهای ارزیابی دخیل باشند.

### ۵-۵- فاز پنجم: تطابق حداکثری با نیازمندی‌ها

پیشنهادات متدولوژی‌های سنتی در مهندسی نرم‌افزار مبنی داشتن تحلیل جامع از کل سیستم قبل از شروع فعالیت پیاده‌سازی از این جهت بوده است که بدلیل بدست آوردن معماری کل سیستم تغییرات در آینده به حداقل کاهش خواهد یافت. این دیدگاه در زمانی صحیح به نظر می‌رسد که نیاز متقاضی محصول نرم‌افزاری در زمان توسعه، تغییر چشمگیری نداشته باشد. به عبارت دیگر متقاضی قبل از شروع فاز طراحی و پیاده‌سازی دقیقاً از نیاز خود مطلع باشد. اما تجربه نشان داده است که در بسیاری از موارد متقاضی یک دید کلی نسبت به نیاز خود دارد و در مورد جزئیات نظری ندارد. به بیان دیگر دقیقاً از نحوه‌ی حل مشکل توسط محصول نرم‌افزاری مطلع نیست. در این حالت است که بر اثر گذشت زمان و قرار گرفتن در جو توسعه‌ی نرم‌افزار از جزئیات مطلع شده و نظرات خود را اعلام می‌دارد. با در نظر گرفتن دلایل اشاره شده، عمل به رویکردهای سنتی در توسعه‌ی نرم‌افزار به دلیل زیاد شدن فاصله‌ی زمانی بین تحلیل و تحویل محصول نرم‌افزاری، منجر به کاهش تطابق محصول تولیدی با نیاز مشتری می‌گردد. لذا در این فاز تصمیم گرفته شد با انتشار نسخه‌های عملیاتی در فواصل زمانی کوتاه و دریافت بازخورد از کاربران شکاف بین نیازمندی‌ها و محصول خروجی را به حداقل کاهش داده شود.

ریسک مهم و قابل توجه در این فاز نداشتن دید جامع نسبت به کل نرم‌افزار در زمان توسعه‌ی نرم‌افزار بود. همانطور که اشاره شد، در رویکرد جدید، فاز ابتدایی تحلیل نرم‌افزار منجر به نگاه جامع در مورد کل سیستم مورد تقاضا نمی‌شد. بلکه بخشی از سیستم مورد هدف قرار می‌گرفت. در این حالت شناخت نیازهای جدید می‌توانست منجر به تغییر زیرساخت‌های تولید شده، ادغام چند زیرساخت با هم و حتی در شرایطی باعث حذف یک زیر ساخت گردد. این تغییرات در بعضی از شرایط می‌توانست هزینه‌ی سنگینی را به تیم توسعه‌دهنده تحمیل نماید. تنها راهی که می‌توانست منجر به کاهش این هزینه‌ها شود، استفاده از افراد با تجربه در بخش طراحی محصول مورد تقاضا بود. یک طراح با تجربه بدلیل داشتن آینده‌نگری می‌توانست زیرساخت‌ها را بگونه‌ای طراحی نماید که در آینده کمترین تغییر در سیستم تحمیل گردد.

## ۵-۶- فاز ششم: رفع انحصار توسعه

در تیم توسعه‌ی نرم‌افزار معمولاً افراد براساس نقش‌هایی مانند تحلیل‌گر، طراح، برنامه‌نویس و ... در کارهای مختلف قرار نمی‌گرفتند. انتصاب افراد معمولاً براساس پروژه‌ها انجام می‌گرفت. به بیان دیگر یک شخص می‌تواند در یک پروژه‌ی نرم‌افزاری با توجه به شرایط زمانی، دارای نقش تحلیل‌گر، طراح و برنامه‌نویس باشد. بنابراین یک سیستم بعد از گذشت زمان به یک یا چند نفر از نظر توسعه و ارتقا وابسته می‌شد. این امر نه تنها قابلیت تصمیم‌گیری را در زمان تخصیص منابع از مدیریت تیم توسعه‌ی نرم‌افزار سلب می‌نمود بلکه رکود در بین افراد تیم را نتیجه می‌داد. زمانیکه اعضای تیم بدلیل انحصاری شدن سیستم‌ها در بین بعضی از افراد تیم قابلیت نمایش توانمندی‌های خویش را نمی‌یافتند، جو رکود در تیم حاکم می‌گشت.

آنچه به عنوان راه حل پیشنهاد گردید، شناورسازی اعضای تیم در پروژه‌های مختلف بود. دو ریسک عمده در راستای دستیابی به این هدف وجود داشت. اولین ریسک عبارت بود از عدم آشنایی اعضای تیم با بخشی از سیستم‌های تولید شده. بنابراین این افراد نمی‌توانستند ادامه‌ی فعالیت‌های آن سیستم را بر عهده بگیرند. بدین منظور تصمیم گرفته شد از تکنیک برنامه‌نویسی دو نفره<sup>۱</sup> استفاده شود. در این تکنیک دو نفر به یک فعالیت تخصیص داده می‌شدند. یکی از این افراد شخصی است که در سیستم مربوطه خبره است و دیگری شخصی است که با سیستم آشنایی ندارد. در این حالت یک نفر در محیط برنامه‌نویسی کد را تولید و دیگری در کنار او کد تولید شده را کنترل می‌نماید. تکنیک دیگری که در این زمینه استفاده شد، برگزاری جلسات طراحی به صورت گروهی بود. در این جلسات برای طراحی یک سیستم یا بخشی از آن تمامی افرادی که باید در آن سیستم دخیل باشند، حضور داشتند. این امر باعث می‌شد که دیدگاه کلی نسبت به سیستم در ذهن افراد ایجاد شود.

دومین ریسک مربوط به هزینه‌ی سربار جابجایی یک شخص بین پروژه‌های مختلف بود. تغییر فضای فکری از یک پروژه به پروژه‌ی دیگر می‌توانست زمان‌بر باشد. راه حلی که برای این ریسک در نظر گرفته شد، کاهش نسبت زمان تغییر فضای فکری نسبت به کل زمان انجام کار جدید بود. به عبارت دیگر در شرایطی جابجایی بین دو کار صورت می‌پذیرفت که زمان انجام کار جدید بیش از یک هفته باشد. در غیر اینصورت کار جدید توسط شخصی انجام می‌شد که در همان فضای فکری در حال انجام فعالیت است.

در جدول ۱ خلاصه‌ای از هر فاز به همراه ریسک‌های مربوطه ارائه شده است.

<sup>1</sup> Pair Programming



جدول ۱- فازهای اجرای تصمیم

فاز	هدف	تکنیک	ریسک
۱	- کاهش تأخیر در انجام وظایف محوله - رفع مشکل زمان‌بندی‌های غیرواقعیانه	- مشارکت تمام اعضای تیم در زمان‌بندی برای آشنایی با نحوه تخمین زمان - ثبت فعالیت‌ها به همراه تخمین زمان برای شفاف‌سازی وضعیت منابع در اختیار - شناورسازی اعضای تیم در کل پروژه‌ها به طوری که بتوان هر فعالیت را به فردی انتصاب داد که کمترین زمان را تخمین زده است.	- عدم آشنایی با نحوه تخمین زمان
۲	مستندسازی و تولید دانش مکتوب جهت جلوگیری از دوباره کاری و خطاهای ناشی از آن	- ثبت نتایج وظایف محوله در هر نيمروز (هر ۳ ساعت یکبار) - تعریف چک لیست انجام کار	- عدم آشنایی با ابزارهای ثبت دانش - زمان‌بر بودن ثبت دانش
۳	ایجاد شفافیت در کارها	- تعریف و پیاده‌سازی فرایندهای انجام کار و پشتیبانی	
۴	ایجاد عامل انگیزش در تیم توسعه نرم‌افزار	- تعریف پارامترهای ارزیابی عملکرد اعضای تیم توسعه نرم‌افزار با استفاده از تکنیک ارائه شده در منبع [۲] - پرداخت پاداش ماهانه به شخص برتر	- عدم رغبت اعضای تیم نسبت به ارزیابی
۵	تطابق حداکثری محصولات با نیازمندی‌های کاربران	- انتشار نسخه‌های عملیاتی در فواصل زمانی کوتاه و دریافت بازخوردهای کاربران - استفاده از افراد با تجربه در بخش طراحی	- هزینه تغییراتی که ناشی از عدم شناخت کل سیستم است
۶	رفع انحصاری شدن بخش‌های مختلف نرم‌افزار برای برخی از اعضای تیم توسعه نرم‌افزار	- شناورسازی اعضای تیم در پروژه‌های مختلف با تکنیک برنامه‌نویسی دو نفره - برگزاری جلسات طراحی گروهی - کاهش نسبت زمان تغییر فضای فکری نسبت به کل زمان انجام کار جدید	- هزینه سربار جابجایی بین پروژه‌ها - عدم آشنایی اعضای تیم با بخشی از سیستم‌های تولید شده

## ۶- نتایج

در طول مدت اجرای فازهای اشاره شده در بخش پیشین، پارامترهایی که برای اندازه‌گیری میزان اثربخش بودن راه‌کار انتخابی تعریف شده بودند، مورد اندازه‌گیری قرار می‌گرفتند. تنظیم کردن راه‌کار انتخابی با شرایط حاکم بر تیم یکی از مهم‌ترین نکات در موفقیت‌آمیز بودن کل راه‌حل است. به عبارت دیگر شرایط حاکم بر تیم در زمان‌هایی باعث می‌شد که یک فاز با تأخیر اجرا شود یا بدلیل شناسایی ریسک‌های جدید با رویکردهای جدیدی پیگیری گردد.

پارامترهایی که برای اندازه‌گیری میزان اثربخشی راه‌کار انتخابی، تعریف شده بودند عبارتند از:

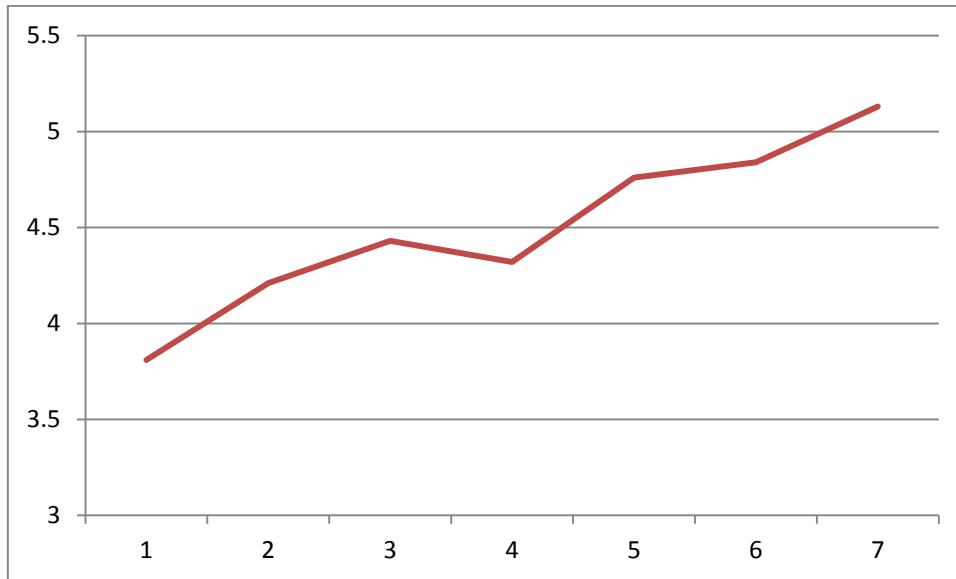
شاخص‌های ارزیابی عملکرد اعضای تیم توسعه نرم‌افزار

میزان تأخیر در انجام تعهدات

میزان رضایت‌مندی متقاضیان

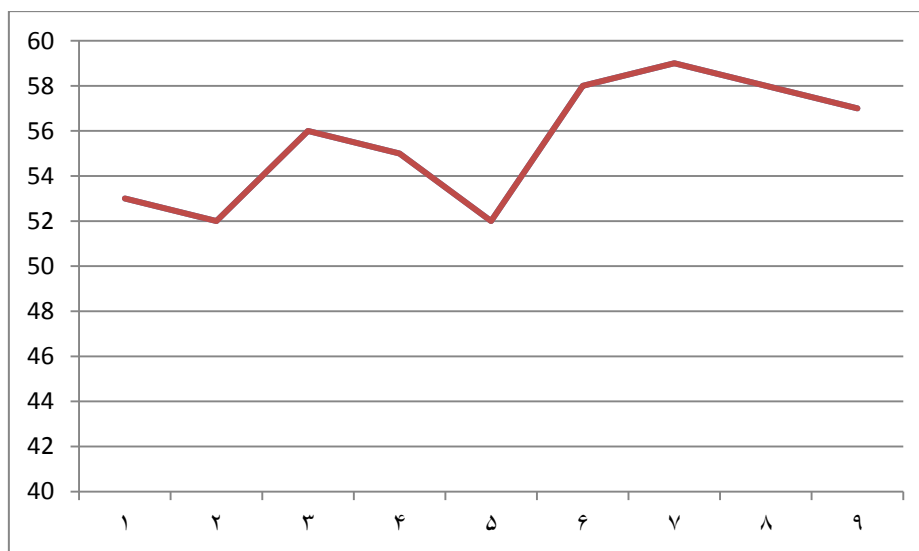
برای اندازه‌گیری اولین پارامتر، ۱۲ فاکتور مختلف که قابل اندازه‌گیری بودند تعریف شده و با نظر کل اعضای تیم از نظر اهمیت وزن‌دهی گردیدند. سپس براساس این فاکتورها، هر ماه کل اعضای تیم ارزیابی شده و رده‌بندی می‌گشتند. همانطور که در شکل ۱ مشاهده می‌شود میانگین امتیاز تیم توسعه رشد قابل قبولی داشته است.





شکل ۱- میانگین امتیاز تیم توسعه‌دهنده در طی ۷ ماه

بررسی‌های صورت گرفته بر روی میزان تأخیر در انجام تعهدات نسبت به آنچه قبل از پیاده‌سازی راه‌کار وجود داشته نشان دهنده قابل قبول بودن تکنیک‌های مورد استفاده است. به صورت میانگین میزان تأخیر در انجام تعهدات به کمتر از ۳۰ درصد کاهش یافته است. البته انتظار می‌رود این مقدار با نهادینه شدن راه‌کارهای پیشنهادی کمتر نیز شود. در نهایت میزان رضایتمندی متقاضیان براساس نظرسنجی‌هایی که ماهانه به صورت تصادفی از تعدادی از کاربران سیستم‌ها صورت می‌پذیرفته، مطابق با شکل ۲ امیدوار کننده است. شیب کم در میزان رضایتمندی متقاضیان می‌تواند حاصل از سابقه‌ای باشد که نسبت به سیستم در خاطر دارند. انتظار می‌رود چنانچه با همین رویکرد، فرایندهای تیم توسعه نرم‌افزار ادامه یابد، میزان رضایتمندی متقاضیان با شیب بیشتری افزایش پیدا کند.



شکل ۲- درصد رضایتمندی متقاضیان در طی ۹ ماه

آنچه به عنوان جمع‌بندی نهایی در این تجربه مدیریتی می‌توان به آن توجه کرد این است که شناخت دقیق از محیطی که به فکر بهینه‌سازی آن هستیم به همراه شناسایی ریسک‌ها و مخاطرات موجود بر سر راه این بهینه‌سازی می‌تواند کمک فراوانی در کامیابی رویکرد پیشنهادی داشته باشد. لذا پایش لحظه به لحظه محیط و گرفتن بازخورد از آن می‌تواند مسیر رویکرد را اصلاح نماید.

## ۷- منابع و مراجع

۱- حسین شریفی‌پناه، بهروز نونهال، ساخت‌یافتگی یا شبیه‌گرایی، کهنه‌پرستی یا ترقی‌خواهی، اولین کنفرانس ملی رویکردهای نوین در مهندسی کامپیوتر و بازاریابی اطلاعات، مهرماه ۱۳۹۲، رودسر.

2- Thompson, Kenneth N. **Scaling evaluative criteria and supplier performance estimates in weighted point prepurchase decision models.** *International Journal of Purchasing and Materials Management* 27.1 (1991): 27-36.