

این کتاب تنها به خاطر حل مشکل دانشجویان پیام نور تبدیل به پی دی اف شد. همین جا از ناشر و نویسنده و تمام کسانی که با افزایش قیمت کتاب ما را مجبور به این کار کردند و یا متحمل ضرر شدند عذرخواهی می کنم. گروهی از دانشجویان مهندسی کامپیوتر مرکز تهران

بخش دوم مدیریت پروژه های نرم افزاری

در این بخش از کتاب مهندسی نرم افزار رهیافتی برای یک اهل فن، شیوه های مدیریت را مورد بحث و بررسی قرار می دهیم که برای طرح ریزی، سازماندهی، نظارت و کنترل پروژه های نرم افزاری مورد نیاز هستند. در فصل های بعدی پاسخ سوالات زیر را می یابیم:

- در طول پروژه نرم افزاری چگونه باید افراد، فرآیند و مسائل را اداره نمود؟
 - معیارهای نرم افزار کدامند و چگونه می توان از آنها برای پروژه و فرآیند نرم افزار استفاده کرد؟
 - تیم یا گروه نرم افزار چگونه برآوردهای معتبر و قابل اطمینانی درمورد کار، هزینه و مدت پروژه تهیه می کند؟
 - برای ارزیابی ریسک هایی که بر موفقیت پروژه ها تأثیر می گذارند، از چه «مهندسی» ای می توان استفاده کرد؟
 - مدیر پروژه نرم افزار، کارکردهای مهندسی نرم افزار را چگونه انتخاب می کند؟
 - برنامه زمان بندی پروژه چگونه خلق می شود؟
 - کیفیت چگونه تشریح می شود تا بتوان آن را کنترل نمود؟
 - تضمین کیفیت نرم افزار چیست؟
 - بررسی های فنی رسمی چرا اینقدر اهمیت دارند؟
 - تغییر در طول توسعه نرم افزار کامپیوتری و بعد از تحویل به مشتری چگونه اداره می شود؟
- زمانیکه به این پرسشها پاسخ دادید می توانید برای اداره کردن پروژه های نرم افزاری به نحوی که به تحویل به موقع محصول با کیفیت بالا بیانجامد آماده شوید.

مفاهیم مدیریت پروژه

فصل ۳

مفاهیم کلیدی (مرتب بر حروف الفبا)

اصول W5HH، اقدامات بحرانی، تجزیه و شکستن مسئله، تیم نرم افزار، چارچوب مشترک فرآیند، حوزه، رهبر تیم، ساختارهایی تیمی، طبقه‌بندی تیم‌ها، هماهنگ سازی

KEY CONCEPTS

critical practices, common process framework, coordination, problem decomposition, process decomposition, scope, software team, team leader, team structure, team toxicity, W5HH principle

نگاه اجمالی

مدیریت چیست؟ البته بسیاری از ما (در لحظات سیاه‌تر و تیره‌تر خود) نظریه ویلبرت را در مورد مدیریت می‌پذیریم. به هنگام ساخته شدن سیستم‌ها و محصولات مبتنی بر کامپیوتر مدیریت همچنان فعالیتی ضروری خواهد بود. مدیریت پروژه با طرح‌ریزی، نظارت و کنترل افراد، فرآیندها و رخدادهایی که با پدیدار شدن نرم‌افزار از مفاهیم اولیه تا اجرای عملی مطرح می‌شوند، مرتبط است.

چه کسی آن را انجام می‌دهد؟ هر کس تاحدی در انجام مدیریت دخیل است اما حیطه فعالیت‌های مدیریتی باتوجه به فردی که آن را به انجام می‌رساند، متغیر خواهد بود. مهندس نرم‌افزار فعالیت‌های روزمره، طرح‌ریزی، نظارت و کنترل کارهای فنی خود را اداره می‌کند. مدیران پروژه کار گروه یا تیم مهندسين نرم‌افزار را طرح ریزی و کنترل می‌کنند. مدیران ارشد ارتباط میان امور و متخصصین نرم‌افزار را هماهنگ می‌کنند.

چرا مدیریت مهم است؟ ساختن نرم‌افزار کامپیوتری کار پیچیده‌ای است خصوصاً اگر افراد بسیاری در آن دخیل باشند و برای مدت نسبتاً طولانی بر روی آن کار کنند. بدین دلیل است که باید پروژه‌های نرم‌افزار را اداره و کنترل کنیم.

مراحل مدیریت کدامند؟ درک کردن چهار P یعنی افراد، محصول، پروسه و پروژه، افراد باید برای انجام درست کار نرم‌افزار سازماندهی شوند. ارتباط با مشتری باید به نحوی باشد که حیطه محصول و نیازمندی‌ها درک شوند. باید پروسه‌ای انتخاب شود که با نرم‌افزار و محصول متناسب باشد. پروژه باید با برآورده کردن نیرو و زمان موردنیاز برای انجام کار، تشریح محصولات، انجام کنترل کیفیت و تعیین مکانیزم‌های نظارت و کنترل کار تشریح شده در طرح، طراحی شود.

محصول چیست؟ طرح پروژه با آغاز و شروع فعالیت‌های مدیریت تهیه و تولید می‌شود. این طرح روندها و کارهایی که باید انجام شوند، افرادی که کارها را انجام می‌دهند و مکانیزم ارزیابی خطرات، کنترل تغییر و ارزیابی کیفیت را تشریح می‌کند.

چگونه می‌توانم مطمئن شوم که کار را درست انجام داده‌ام؟ هرگز نمی‌توانید بطور قطعی اطمینان حاصل کنید که طرح پروژه درست است مگر آنکه محصول با کیفیت بالا را در زمان مقرر و با بودجه تعیین شده ارائه کنید. البته مدیر پروژه به هنگام ترغیب پرسنل نرم‌افزار در کار کردن بصورت یک گروه و تأکید بر توجه آنها به نیازهای مشتری و کیفیت محصول مدیریت درست را اجرا می‌کند.

میلریج جونز [PAG85]^۱ در پیش‌گفتار کتاب مدیریت پروژه نرم‌افزار خود مطالبی را بیان کرد که بسیاری از مشاورین مهندسی نرم‌افزار می‌توانند آنها را بازگو کنند:

«من کارگاه‌های تجاری چندی را، خوب و بد، ملاحظه کرده‌ام و مدیران پردازش داده‌های خوب و بد را نیز مشاهده نموده‌ام. اغلب با وحشت به این چیزها نگریده‌ام زیرا، این مدیران به گونه‌ای بیهوده در پروژه‌هایی کابوس مانند، با تاریخ‌های سررسید غیرممکن یا سیستم‌هایی که کاربران آنها را به هیچ می‌انگارند، به تلاش عبث پرداخته‌اند و می‌کوشند تا بخش‌های بزرگی از زمان نگهداری را با حرص و ولع بکار گیرند و به هدر دهند.»

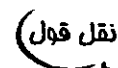
آنچه بیج جونز تشریح می‌کند عوارضی هستند که با مجموعه مسائل فنی و مدیریتی ناشی می‌شود. البته اگر قرار بود هر پروژه‌ای را پیش از پایان آن تشریح و بررسی کنیم به احتمال زیاد با موضوعاتی مشابه و یکسان مواجه می‌شدیم یعنی مدیریت ضعیف.

در این فصل و شش فصل بعدی مفاهیمی کلیدی را بررسی خواهیم کرد که به مدیریت کارآمد پروژه نرم‌افزاری می‌انجامد. در این فصل مفاهیم و اصول اولیه مدیریت پروژه نرم‌افزاری را بررسی می‌کنیم. در فصل ۴ روند و نحوه سنجش پروژه مطرح می‌شود که بنایی برای تصمیم‌گیری مدیریت کارآمد است. شیوه‌های مورد استفاده برای برآورد و تخمین هزینه و نیازمندی منابع و برقراری طرح مؤثر پروژه در فصل ۵ تشریح و بررسی می‌شوند.

فعالیت‌های مدیریت که به مشخص کردن و تخفیف یا کاهش مؤثر و مدیریت خطرهای می‌انجامد، در فصل ۶ مطرح می‌شوند. در فصل ۷ فعالیت‌های موردنیاز برای تشریح وظایف و کارکردهای پروژه و ارائه زمانبندی قابل اجرا پروژه بررسی می‌شوند. در نهایت فصل‌های ۸ و ۹ شیوه‌های تضمین کیفیت در زمان انجام پروژه و کنترل تغییرات در طول آن، در نظر گرفته شده‌اند.

۳-۱ طیف مدیریت

مدیریت پروژه نرم‌افزار کارآمد بر چهار P (مخفف کلمات کلیدی - م) تأکید دارد: افراد^۱، محصول^۲، پروسه^۳ و پروژه^۴، ترتیب این موارد اختیاری نیست. مدیری که فراموش می‌کند که کار مهندس نرم‌افزار تلاشی انسانی است هرگز نمی‌تواند در مدیریت پروژه موفق باشد. مدیری که نمی‌تواند در پیشرفت و تکامل پروژه ارتباط با مصرف‌کننده را ترغیب کند پاسخ و راه حل خوبی را برای یک مسئله غلط می‌یابد. مدیریت که بر پروسه و فرآیند، کم توجهی می‌کند خطر اعمال روشهای فنی و ابزار مناسب در جای غیرضروری را به جان می‌خرد. مدیری که بدون طرح پروژه شروع به کار می‌کند موفقیت محصول را به مخاطره می‌اندازد.



قابلیت متفاوت افراد در انجام وظائف برنامه‌سازی موجب تغییرات بسیار شگفتی در کار خواهد بود. بیل کورنیس

۳-۱-۱ افراد

پرورش پرسنل نرم‌افزار کاملاً ماهر و با انگیزه از دهه ۱۹۶۰ به بعد مورد بحث قرار گرفته است (مثلاً [COU80]^۵ و [WIT94]^۶ و [DEM98]^۷)، درواقع فاکتور انسانی (افراد یا پرسنل) آنقدر مهم است که مؤسسه مهندسی نرم‌افزار برای بالابردن آمادگی سازمان‌های نرم‌افزاری در اجراء کاربردهای پیچیده با کمک کردن به جذب، پرورش، ترغیب، بکارگیری و حفظ قابلیت‌های موردنیاز برای بهبود (قابلیت‌های توسعه نرم‌افزاری افراد مدل رشد قابلیت مدیریت افراد (PM-CMM)^۸ را ارائه کرده است. مدل رشد مدیریت افراد عملکردهای کلیدی زیر را برای پرسنل نرم‌افزار تشریح می‌کند: استخدام، گزینش، مدیریت عملکرد، آموزش، جبران، توسعه و پیشرفت شغلی، طراحی سازمان و کار و توسعه تیم/فرهنگ. [CUR94]^۹



یک طبقه بندی از حوزه های کاربردی که تولید نرم افزار در آنها صورت می پذیرد. در فصل ۱ آورده شده است.

PM-CMM عامل ملازم و همراه مدل رشد قابلیت نرم‌افزار (فصل ۲) است که سازمان‌ها را در خلق روند نرم‌افزار هدایت می‌کند. مسائل مربوط به مدیریت افراد و ساختار پروژه‌های نرم‌افزار در بخش‌های بعدی این فصل بررسی می‌شوند.

1. People
2. Product
3. Process
4. Project
5. Cougar, J. and R.
6. Whitaker, K.
7. DeMarco, T.
8. People management capability maturity model
9. Curtis, B.

۲-۱-۳ محصول

قبل از طرح ریزی پروژه باید اهداف و حیطه محصول^۱ مشخص شده، راه حل های مختلف در نظر گرفته شده و محدودیتها و قیدهای فنی و مدیریتی تشریح شوند. بدون این اطلاعات نمی توان برآوردهای معقولی از هزینه، ارزیابی مؤثر خدمات، تجزیه و تحلیل کارها و وظائف پروژه یا برنامه زمانبندی قابل اجراء پروژه همراه با نشانه های درست پیشرفت را ارائه کرد.

ارائه کننده نرم افزار و مشتری باید اهداف و حیطه محصول را مشخص کنند. در بسیاری از موارد این فعالیت بصورت بخشی از مهندسی سیستم یا مهندسی روند کار شروع می شود (فصل ۱۰) و به عنوان مرحله لول در تحلیل نیازمندی های نرم افزار ادامه پیدا می کند (فصل ۱۱). اهداف، مقاصد کلی محصول را (از نقطه نظر مشتری) بدون در نظر گرفتن نحوه حصول بر این اهداف تعیین می کند. حیطه محصول^۲ داده ها، عملیات و رفتارهای اولیه مشخص کننده محصول را تشریح می کند و مهمتر از آن می گوید تا به شکلی کمی، این ویژگی ها را با یکدیگر مرتبط کند. بعد از درک اهداف و حیطه محصول، راه حل های مختلف در نظر گرفته می شوند.

۲-۱-۳ فرآیند

پروژه یا روند نرم افزار (فصل ۲) چهارچوبی را ارائه می کند که به کمک آن می توان طرح جامعی جهت ارائه نرم افزار تهیه کرد. تعداد کمی از فعالیت های کاری در تمام پروژه های نرم افزاری - بدون توجه به اندازه یا پیچیدگی آنها - قابل اجراء هستند. مجموعه های مختلف کاری - وظایف، مراحل برجسته کاری، محصولات کاری و مقاطع تضمین کیفیت - به فعالیت های اصلی امکان می دهند تا در مشخصه های پروژه نرم افزار و نیازمندی های تیم پروژه بکار گرفته شوند. در نهایت فعالیت های حفاظتی مثل تضمین کیفیت نرم افزار، مدیریت پیکربندی نرم افزار و اندازه گیری یا سنجش برمدل فرآیند حاکم هستند. فعالیت های حفاظتی از فعالیت های اصلی مستقل هستند و در کل پروژه رخ می دهند.

۴-۱-۳ پروژه

ما پروژه های نرم افزاری کنترل و طرح ریزی شده را به یک دلیل اصلی انجام می دهیم زیرا این تنها راه شناخته شده برای اداره کردن و کنترل پیچیدگی است و هنوز هم با آن در کشمکش و جدال هستیم. در سال ۱۹۹۸ داده های صنعتی نشان داد که ۲۶ درصد از پروژه های نرم افزاری بلافاصله شکست خوردند و



وظائف، معیارها،

محصول و نکات

تضمین کیفیت،

فعالیت های چارچوبی را

به گونه ای مقبول

تبدیل خواهد نمود.

۱. در این متن، واژه محصول به هر نرم افزاری که کسی درخواست ساخت آن را داده است، اطلاق می شود. بنابراین نه تنها مشتمل بر محصولات نرم افزاری خواهد بود که سیستم های مبتنی بر کامپیوتر، نرم افزارهای جاسازی شده، نرم افزارهای حل المسائل (مانند برنامه های حل مسائل علمی / مهندسی) را نیز شامل خواهد شد.

۴۶ درصد از آنها با گذشتن از حدود تعیین شده هزینه و برنامه زمانبندی شده مواجه شدند [REE99]. البته نرخ موفقیت پروژه‌های نرم افزاری ناحدی بهبود پیدا کرده است اما نرخ شکست و عدم موفقیت پروژه همچنان بالاتر از حدی است که باید باشد.^۲

برای پرهیز از شکست پروژه، مدیر پروژه نرم‌افزار و مهندسین نرم افزاری که محصول را می‌سازند باید از علائم اخطار دهنده پرهیز کنند، عوامل موفقیت را که به مدیریت پروژه خوب می‌انجامد درک کنند و روش مشترک و معقولی برای طرح ریزی، نظارت و کنترل پروژه ارائه کنند. هر کدام از این موضوع ها در بخش ۳-۵ و فصل‌های بعدی مورد بحث قرار خواهند گرفت.

نقل قول

شرکتهایی که
مدیرانشان نسبت به
پیشرفت کارکنان
حساس می باشند، در
دراز مدت وضعیت
مناسبی خواهند
یافت.
تام دمارکو و تیم
لیستر

۲-۳ افراد

در تحقیق انتشار یافته توسط [CUR88] IEEE^۱ از نایب رؤسای مهندسی سه شرکت فنی بزرگ خواسته شد تا مهمترین عامل دخیل در پروژه نرم‌افزار موفق را ذکر کنند. پاسخ آنها به شرح زیر بود:
نایب رئیس اول: فکر می‌کنم اگر بخواهیم یک چیز را به عنوان مهمترین عامل در محیط خود ذکر کنیم آن چیز ابزار مورد استفاده ما نیست بلکه افراد یا پرسنل است.

نایب رئیس دوم: مهمترین عاملی که باعث موفقیت این پروژه شد برخورداری از پرسنل باهوش بود ... دیگر عوامل اهمیت کمتری دارند ... مهمترین کاری که درمورد یک پروژه انجام می‌دهید انتخاب کارکنان است ... موفقیت سازمان عرضه کننده نرم‌افزار ارتباط بسیار زیادی با قابلیت بکارگیری و استخدام افراد خوب دارد.

نایب رئیس سوم: تنها قاعده‌ای که در مدیریت از آن استفاده می‌کنم تضمین در اختیار داشتن افراد خوب - افراد واقعاً خوب - و پرورش افراد خوب و ایجاد محیطی است که در آن بتوان افراد خوبی درست کرد.

درواقع این مطالب تأثیری است بر اهمیت افراد در روند مهندسی نرم‌افزار و هنوز همه ما از نایب رؤسای مهندسی ارشد گرفته تا کارشناس پائین‌ترین سطوح، اغلب افراد را به واسطه توانایی‌هایشان برمی‌گزینیم. مدیران (همانند افراد فوق) می‌گویند که پرسنل عامل اصلی هستند اما گاه کردار و عملکرد آنها با گفتارشان یکی نیست. در این بخش افرادی را بررسی می‌کنیم که در روند نرم‌افزار مشارکت دارند و نحوه سازماندهی آنان برای انجام کارآمد مهندسی نرم‌افزار را نیز بررسی می‌کنیم.

1. Reel, J.S.

۲. با این آمارها منطقی است که بپرسیم چگونه رشد نمایی کامپیوترها و صنایع نرم افزاری هنوز ادامه دارد و حتی دو رقیبی است؟ بخشی از پاسخ آن است که، به گمان من، تعداد معتنابهی از این پروژه ها ی شکست خورده، در مکان هایی بوده اند که برای اولین بار پروژه انجام می شده. مشتریان علاقه خود را سریعاً از دست می دادند (زیرا آنچه متقاضی آن بودند، با آنچه واقعا وجود داشت، مطلقاً همخوانی نداشت). و لذا پروژه ها متوقف می شدند.

۱-۲-۳ بازیگران

بازیگران صحنه پروسه نرم افزار (و هر پروژه نرم افزار) را می توان در یکی از پنج گروه زیر طبقه بندی کرد:

۱- مدیران ارشد که موضوعات و مسائل کاری را مشخص می کنند که اغلب تأثیر زیادی بر پروژه دارند.

۲- مدیران پروژه (فنی) که باید افرادی را که کار نرم افزار را انجام می دهند سازماندهی، کنترل و نظارت کنند.

۳- متخصصین افرادی که مهارت های فنی مورد نیاز جهت مهندسی محصول یا کاربرد ارائه می کنند.

۴- مشتریان، کسانی که نیازمندی های نرم افزار مورد نیاز را مشخص می کنند و سهامدارانی که در پی آمد آن نفعی دارند.

۵- کاربران نهایی که با عرضه نرم افزار جهت استفاده از آن بهره می گیرند.

تیم پروژه برای کارآمد بودن باید به نحوی سازماندهی شود که مهارت ها و قابلیت های هر فرد را به بیشترین حد ممکن برساند. و این وظیفه رهبر تیم است.

۲-۲-۳ رهبران تیم

مدیریت پروژه فعالیتی در تعامل با افراد است و به این دلیل کاورزان ذیصلاح اغلب رهبران تیم خوبی نیستند. آنان از مجموعه مهارت های فردی درست و لازم برخوردار نیستند. و همانطوریکه اجمون می گوید «متأسفانه و درغالب موارد به نظر می آید که افراد فقط با قرار گرفتن در موقعیت مدیر پروژه بصورت تصادفی به مدیران پروژه تبدیل می شوند» [EDG95].^۱

جری وینبرگ [WEI86]^۲ در کتاب ارزشمند رهبری فنی، یک مدل MOI^۳ برای رهبری ذکر می کند:

انگیزه، قابلیت ترغیب پرسنل فنی در ایجاد بهترین قابلیت ها.

سازماندهی، توانایی ایجاد و برقراری پروسه های موجود (یا ابداع پروسه های جدید) که امکان تبدیل مفهوم اولیه به محصول نهایی را فراهم می کند.



هنگامیکه کسی را

برای هدایت پروژه نرم

افزاری انتخاب می

کنیم، به دنبال چه می

باشیم؟

1. Curtis, B.

2. Edgemon, J.

3. Weinberg, G.

4. Motivation - Organization - Idea (Innovation)

ایده‌ها یا ابداع و نوآوری. توانایی ترغیب افراد به خلاقیت و احساس خلاقیت به هنگام کار در محدوده تعیین شده برای محصول یا کاربرد نرم افزاری مشخص.

وین برگ می‌گوید که رهبران پروژه موفق سبک مدیریت حل مسئله‌ای اعمال می‌کنند. یعنی مدیر پروژه نرم‌افزار باید بر درک مسئله‌ای که باید حل شود، اداره کردن جریان ایده‌ها و مجاز کردن افراد تیم به آگاهی و اطلاع از موارد کیفی و عدم مصالحه تأکید داشته باشد.

نظریه دیگری [EDG95]^۱ که مدیر پروژه کارآمد را تشریح می‌کند بر چهار ویژگی کلیدی تأکید دارد:

حل مسئله. مدیر پروژه نرم‌افزار کارآمد می‌تواند مسائل فنی و سازمانی شایع را تشخیص داده، راه حل اصولی سازماندهی کند یا دیگر کارورزان را به ارائه راه حل ترغیب کند، مطالب اندوخته شده از پروژه‌های قبلی را در موقعیت‌های جدید اعمال کند و در صورت بی‌ثمر بودن راه حل‌های اولیه شبکه برای تغییر جهت از انعطاف‌پذیری کافی برخوردار باشد.

هویت مدیریتی. مدیر پروژه خوب باید مسئولیت پروژه را برعهده بگیرد. او باید برای بدست گرفتن کنترل در زمان مقتضی و مجاز کردن پرسنل فنی خوب به دنبال کردن غرائز خود از اعتماد (اتکاء به نفس) کافی برخوردار باشد.

موفقیت. برای بهینه کردن بهره‌وری تیم پروژه، مدیر باید نوآوری و موفقیت را تشویق و ترغیب کند و با اعمال خود نشان دهد که اگر افرادی به طور کنترل شده ریسک کنند، با مجازات مواجه نخواهند شد.

ساخت تیم و تاثیرگذاری. یک مدیر موفق و موثر باید مردم شناس باشد، سیگنال‌های کلامی و غیر گفتاری را بفهمد، به آنها واکنش مناسب نشان دهد. مدیر باید تحت شرایط فشار بتواند اوضاع را تحت کنترل داشته باشد.

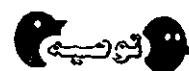
۳-۲-۳ تیم نرم‌افزار

تعداد ساختارهای سازمانی انسانی برای توسعه نرم‌افزار، به اندازه تعداد سازمانهای توسعه دهنده نرم افزاری می‌باشد. ساختار سازمانی نمی‌تواند به راحتی تغییر داده شود و این مسأله ممکن است خوب یا بد باشد. نتایج عملی و سیاسی تغییر سازمانی در حیطه مسئولیتهای مدیر پروژه نرم‌افزاری نمی‌باشد. اما تشکیلات مردمی که مستقیماً در پروژه جدید نرم‌افزاری دخیل هستند در حیطه کاری مدیر پروژه قرار دارد.

اختیارات ذیل برای بکار بردن منابع انسانی در یک پروژه که به n نفر نیروی کار برای k سال نیاز دارد، موجود می‌باشد.

نقل قول

بیان ساده. یک
بسر آن است که می
د به کجا باید برود.
بر می‌خیزد و می‌رود.
جان لرسکین



یک جادوگر نرم‌افزار.
یک خبره نرم‌افزار،
محمّل است که
نخواهد یا نتواند که
رهبری یک تیم را
عهده دار شود. او را به
این کار وادار نکنید.

نقل قول

نه هر گروهی یک تیم
است و نه هر تیمی
یک تیم موفق.
کلن بارکر

۱- تعداد n نفر نیروی کار برای تعداد m کار عملی مختلف تعیین می شوند که در این کارها میزان کار مشترک نسبتاً کم می باشد؛ هماهنگی کار، جزء مسئولیت مدیر نرم افزار می باشد که ممکن است در ۶ پروژه دیگر نیز مشغول باشد.

۲- تعداد n نفر برای کار در تعداد m کار عملی مختلف ($m < n$) طوری تعیین می شوند که «تیم های» غیررسمی ایجاد می شوند. و یک رئیس متخصص تیم نیز ممکن است در این دوره مشخص شود. ایجاد هماهنگی میان تیم برعهده مدیر نرم افزار است.

۳- تعداد n نفر نیرو کار در t تیم تشکیل می شوند و برای هر یک از تیمها یک یا چند کار عملی تعیین می شود؛ هر یک از تیمها دارای ساختار خاصی می باشند که برای تمام تیمهایی که در این پروژه مشغول هستند تعریف شده می باشد؛ و هماهنگی بوسیله مدیر پروژه نرم افزاری و مدیر تیم کنترل می شود.

اگرچه می توان درمورد هر یک از رهیافتهای بالا بحث هایی را ترتیب داد که به موافقت یا مخالفت با این رهیافتها بپردازد، اما شواهد رو به رشدی وجود دارد که نشان می دهد یک تشکیلات تیم رسمی سودمندترین و بهره ورترین رهیافت می باشد.

«بهترین» ساختار تیمی به سبک مدیریتی سازمان شما، تعداد افراد تشکیل دهنده تیم و سطح مهارت آنها و سختی مسائل بستگی دارد. مانتی [MAN81]^۱ سه سازمان کلی تیمی را پیشنهاد می کند. این سازمانها عبارتند از:

سازمان غیرمتمرکز دموکراتیک (DD):^۲

این تیم مهندسی نرم افزاری دارای یک رئیس دائمی و ثابت نمی باشد. بلکه «هماهنگ کننده های کار برای دوره کوتاهی در این سمت منصوب می شوند و سپس افراد دیگری که ممکن است کارهای مختلفی را هماهنگ کنند جایگزین این هماهنگ کننده ها می شوند.» تصمیم گیری درمورد مشکلات و رهیافت مورد نظر بوسیله اتفاق آراء در گروه انجام می شود. ارتباطات موجود میان اعضاء تیم به صورت هم سطح (افقی) می باشد.

سازمان غیر متمرکز کنترل شده (CD):^۳

این تیم مهندسی نرم افزاری دارای یک رئیس مشخص می باشد که کارهای خاص و مدیران ثانوی که مسئولیت کارهای فرعی را برعهده دارند، هماهنگ و مشخص می کند. اما حل مشکلات به صورت یک کار گروهی باقی می ماند، اما اجرای راه حلهای مشخص شده برای مشکل توسط رئیس تیم در میان



ک تیم نرم افزاری
گونه باید سازمان
بد؟

1. Mantei, M.

2. Democratic decentralized

3. Controlled decentralized

گروه‌های مختلف تقسیم می‌شود. ارتباطات موجود میان گروه‌های فرعی و افراد به صورت هم سطح (افقی) می‌باشد. همچنین ارتباط عمودی در امتداد سلسله مراتب کنترل بوقوع می‌پیوندد.

سازمان متمرکز کنترل شده (CC):^۱

حل مشکلات در سطوح بالا و ایجاد هماهنگی داخلی در میان تیم تحت نظارت و کنترل رئیس تیم انجام می‌شود. ارتباط موجود میان رئیس تیم و اعضای تیم به صورت عمودی می‌باشد.

مانتی [MAN81]^۲ همچنین هفت فاکتور پروژه‌ای که باید در هنگام برنامه‌ریزی ساختار تیم‌های مهندسی نرم افزاری در نظر گرفته شوند را به شرح زیر توصیف می‌کند:

- سختی مشکلاتی که باید مورد حل و فصل قرار گیرند.
- اندازه برنامه (های) حاصله در تعداد خطوط دستورالعملها و جملات برنامه کامپیوتری یا امتیازات کارکردی (فصل ۴).
- مدت زمانی که تیم در کنار یکدیگر باقی خواهند ماند (مدت اشتغال تیم).
- تاجه اندازه می‌توان مشکلات را به صورت پیمانه ای درآورد.
- میزان اعتبار و کیفیت موردنیازی که در ارتباط با سیستم باید ایجاد شود.
- ثبات تاریخ تحویل.
- میزان جامعه پذیری (ارتباطات) موردنیاز برای پروژه.

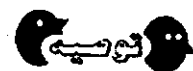
از آنجایی که ساختار متمرکز کارها را سریعتر تکمیل می‌کند، بنابراین این ساختار در حل مشکلات ساده مفیدترین ساختار می‌باشد. تیم‌های غیرمتمرکز نسبت به افراد راه‌حلهای بیشتر و بهتری را بوجود می‌آورند. بنابراین احتمال موفقیت چنین تیم‌هایی هنگام بررسی مسائل بسیار سخت بیشتر می‌باشد. از آنجایی که تیم متمرکز کنترل شده (CD) برای حل مشکلات به صورت متمرکز عمل می‌کند، بنابراین ساختارهای CD و CC هر دو می‌توانند در امر حل مشکلات ساده به صورت موفق عمل کنند. و ساختار DD برای حل مشکلات سخت و پیچیده بهترین ساختار می‌باشد.

از آنجایی که عملکرد تیم با میزان ارتباطاتی که باید میان تیم وجود داشته باشد رابطه (نسبت) معکوس دارد، بنابراین هنگامی که گروه‌های فرعی را بتوان براحتی تطبیق داد، پروژه‌های بسیار بزرگ نیز به بهترین نحو توسط تیم‌هایی که دارای ساختارهای CC یا CD هستند مورد بررسی قرار خواهد گرفت. طول مدت زمانی که اعضای تیم در کنار یکدیگر خواهند بود بر خصوصیات اخلاقی تیم تأثیرگذار خواهد بود. مشخص شده است که در ساختارهای نیمه DD رضایت اخلاقی و کاری به میزان بسیار بالایی



چه عوامل و

فاکتورهایی در تعیین
ساختارهای یک تیم
نرم افزاری، باید مد
نظرمان باشد؟



اغلب بهتر است که تیم
های کوچک ولی
تخصصی داشته باشیم.
تا یک تیم بزرگ.

1. Controlled Centralized

2. Mantei, M.

وجود آمده است و بنابراین کارکردن افراد تیم در کنار یکدیگر در مدت زمان طولانی بسیار خوب خواهد بود.

ساختار تیم DD درمورد حل مشکلاتی که بیمانه های مجزای آن نسبتاً کم می باشد به خوبی عمل خواهد کرد به دلیل آنکه در این نوع ساختار میزان ارتباطات بیشتری مورد نیاز می باشد. هنگامی که حالت بیمانه ای به میزان زیادی وجود داشته باشد ممکن است (افراد تیم بتوانند کار خود را انجام دهند)، ساختار CC یا CD به خوبی عمل کند.

مشخص شده است که تیم های CC و CD در مقایسه با تیم های DD کاستی های کمتری را ایجاد می کنند. اما این داده ها بیشتر با فعالیتهای خاص تضمین کیفیت که بوسیله تیم بکار برده می شود در ارتباط می باشند. بطور کلی تیم های غیر متمرکز نسبت به ساختار متمرکز مدت زمان بیشتری را برای تکمیل یک پروژه نیاز دارند و در عین حال هنگامی که ارتباطات (جامعه پذیری) بیشتری مورد نیاز است این تیم های غیر متمرکز هستند که بهترین ساختار را تشکیل می دهند.

کنستانتین [CON93]^۱ چهار «الگوی سازمانی»^۲ را برای تیم های مهندسی نرم افزاری پیشنهاد کرده است. این پارادایم ها عبارتند از:

۱- الگوی بسته^۳: این نمونه، تیم مورد نظر را با توجه به سلسله مراتب سنتی قدرت و اختیارات تشکیل می دهد. (مشابه با تیم های CC). چنین تیم هایی هنگام تولید نرم افزاری که کاملاً مشابه با تلاشهای قبلی است به خوبی عمل می کنند، اما احتمال خلاق بودن چنین تیم هایی هنگام کارکردن در یک نمونه بسته بسیار کم می باشد.

۲- الگوی تصادفی^۴: تیم را به صورت تصادفی و آزادانه تشکیل می دهد و به ابتکار فردی اعضا تیم بستگی دارد. هنگامی که تحول ابداعات یا فناوریها مورد نیاز باشد، تیمی که دارای نمونه تصادفی است به خوبی عمل خواهد کرد. اما چنین تیمی در صورتیکه «عملکرد منظم و مرتب» مورد نیاز باشد با مشکل مواجه خواهد شد.

۳- الگوی باز^۵: این نمونه سعی دارد تیمی را به روشی تشکیل دهد که این تیم دارای بعضی از کنترل های مربوطه نمونه بسته باشد و در عین حال بیشتر ابداعات پدید آمده در صورت استفاده از نمونه تصادفی را نیز دارا باشد. کار در این نمونه به صورت دسته جمعی و همراه با ارتباط بیشتر و توافق آرای مبتنی بر تصمیم گیری و علائم تجاری تیم های نمونه باز انجام می شود. ساختارهای تیم

نقل قول

تمامی و کار با افراد
مشکل است اما غیر
ممکن نیست.
پیتر دروکر



نقش افرادی که در
فرایند نرم افزار حکم
کتابدار را دارند.
معمولاً غیر مرتبط با
ساختار تیمی است.

1. Constantine, L.

۲. Organizational Paradigms

3. closed paradigm

4. random paradigm

5. open paradigm

نمونه باز برای یافتن راه حل برای مشکلات پیچیده بسیار مناسب هستند. اما برای حل این مشکلات به اندازه سایر تیم‌ها سودمند نمی‌باشند.

۴- الگوی همگام^۱: این نمونه بر تشابه طبیعی مشکل متکی می‌باشد و اعضای تیم را برای کار

کردن بر روی بخشهایی از مشکل سازماندهی می‌کند و در طول این کار ارتباط فعال میان اعضاء تیم بسیار کم می‌باشد.

تغییر در تیم غیر متمرکز دموکراتیک از کنستانتین [CON93]^۲ پیشنهاد شده است. وی طرفدار تیم‌هایی است که دارای استقلال خلاق می‌باشند و بهترین نام برای رهیافت کاری این تیم‌ها «بی‌نظمی ابتدایی»^۳ می‌باشد. اگرچه رهیافت آزاد برای کار نرم افزاری جذاب و جالب است، اما «مهار کردن هرج و مرج موجود در تیمی با عملکرد بالا» باید هدف مرکزی و اصلی سازمان مهندسی باشد. برای دستیابی به تیمی با عملکرد عالی و بالا باید مولرد زیر رعایت شود:

- اعضاء تیم باید به یکدیگر اعتماد داشته باشند.
- توزیع مهارتها باید متناسب با مشکل موجود باشد.
- برای حفظ انسجام و یکپارچگی تیم، افراد مستقل از تیم و خودسر باید از تیم حذف شوند.
- صرف نظر از سازمان تیم، هدف اصلی هر مدیر پروژه‌ای کمک به تشکیل شدن تیمی است که دارای انسجام و یکپارچگی می‌باشد. دمارکو و لیستر [DEM98]^۴ در کتاب خود با نام «فرد افزار»^۵ به بحث در مورد این مطلب می‌پردازد:

ما لغت تیم را استفاده می‌کنیم تا آزادی نسبی را در دنیای اقتصاد نشان دهیم. و هر گروه از افراد را که برای کار کردن با یکدیگر تعیین شده اند یک «تیم» می‌نامیم. اما بسیاری از این گروه‌ها شباهتی به تیم ندارند. آنها تعریف مشترکی از موفقیت یا روحیه تیمی قابل تشخیص و متمایزی ندارند. اما آنچه که در این تیمها وجود ندارد پدیده‌ای است که ما «همبستگی» می‌نامیم.

یک تیم همبسته شامل گروهی از افراد است که به شدت به یکدیگر پیوند خورده‌اند بطوریکه کل این تیم بزرگتر از مجموع بخشهای آن می‌باشد

هنگامی که همبستگی میان یک تیم آغاز می‌شود، احتمال موفقیت افزایش می‌یابد. در این حالت نمی‌توان تیم را متوقف کرد و موفقیت حاصل شده در بین تیم همچنان ادامه دارد. و لازم نیست که این گروه از افراد به روش سنتی تحت کنترل قرار گیرند و مطمئناً آنها نیازی به تشویق شدن و تحریک شدن ندارند. آنها نیروی لازم برای تحرک و سوق دادن را دارا می‌باشند.

نقل قول

مهم نیست که مشکل کدام است. مشکل از ماست (افراد). جری وین برگ

1. synchronous paradigm

2. Constantine, L.

3. innovative anarchy

4. DeMarco, T and T.

6. People ware.

هماکرو ولیستر اثبات کرده اند که اعضای تیم های همبسته بیشتر از حد متوسط سودمند هستند و انگیزه آنها برای انجام کار بیش از حد متوسط می باشد. اعضای این تیم دارای هدف مشترک، فرهنگ مشترک می باشند و در بسیاری از موارد «حس مشترک نخبه بودن» در میان اعضای این گروه موجب شده است که آنها به صورت گروه بی نظیر و منحصر بفردی در آیند.

اما تمام تیمها به صورت همبسته نمی باشند. و در واقع بسیاری از تیمها از نبود این همبستگی که جکمن آن را «مسمومیت تیمی» [JAC98]^۱ می نامد، رنج می برند. وی پنج فاکتور را تعریف می کند که این فاکتورها «موجب پیشرفت مسمومیت در محیط تیم می شوند». این فاکتورها عبارتند از:

۱- یک جوکاری آشفته که در آن اعضای تیم انرژی را اتلاف می کنند و بر اهداف لازم الاجرا کار توجه و تأکیدی ندارند.

۲- ناامیدی شدید که بوسیله فاکتورهای پرسنل، کار یا فناوری ایجاد می شود و این ناامیدی میان اعضای تیم اختلاف ایجاد می کند.

۳- شیوه های تفکیک شده یا ناقص هماهنگی یا تعریف ناقص یا انتخاب نادرست، مدلی را ایجاد می کند که مانعی برای تکمیل کار خواهد بود.

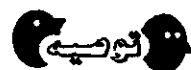
۴- تعریف نامشخص نقش ها (نقش کارکنان) موجب از بین رفتن حس مسئولیت می شود.

۵- تعریف قرارگیری در معرض شکست (ناکامی) مکرر و پیوسته که موجب از بین رفتن اعتماد به نفس و تنزل مسائل اخلاقی می شود.

جکمن برای مشکلات فوق الذکر و سایر مشکلات رایج در این زمینه کارهایی را پیشنهاد کرده است که این سموم را دفع می کنند.

برای اجتناب از یک محیط کاری آشفته، مدیر طرح باید مطمئن شود که تمام اطلاعات مورد نیاز برای انجام کار و تمام اهداف و مقاصد اصلی کار دسترسی دارد و در اهداف تعریف شده کار نباید تغییری ایجاد شود، مگر در صورت لزوم. علاوه بر آن خبرهای بد نباید به صورت راز باقی بماند بلکه باید این خبرها تا حد ممکن به سرعت به سایر اعضای گروه نیز گفته شود (درحالیکه هنوز زمان برای نشان دادن عکس العمل به روش منطقی و کنترل شده وجود دارد).

اگر چه ناامیدی دلایل بسیاری دارد، اما کارکنان نرم افزاری اغلب آن را هنگامی تجربه می کنند که آنها برای کنترل وضعیت خود قدرتی در خود نمی بینند. یک تیم نرم افزاری در صورتی می تواند از ناامیدی بدور باشد که تا حد ممکن مسئولیتهایی در زمینه تصمیم گیری به این تیم واگذار شود. هرچه قدر کنترل روند و تصمیم گیریهای فنی بیشتر به تیم واگذار شود، اعضای تیم کمتر احساس ناامیدی و عجز خواهند کرد.



بلشتن تیمهای زل (که اعضای آن با تمطاف پذیری کامل، به بلوغ همفکری رسیده اند) ایده آل است. اما دسترسی به این مهم آسان نیست. حداقل به این اطمینان برسید که محیط "مسموم" نمی باشد.



چگونه از "مسمومیت" که اغلب تهدید کننده تیم های نرم افزاری است، اجتناب کنیم؟

می‌توان از انتخاب نامناسب روند نرم افزاری (به عنوان مثال کارهای غیرضروری و سنگین و ناگوار یا انتخاب نامناسب محصول کاری) به دو طریق اجتناب نمود. این دو روش عبارتند از:

۱- مطمئن شدن از این امر که مشخصات نرم‌افزار مورد ساخت مطابق با سختی و عدم انعطاف روند انتخاب شده می‌باشد یا خیر.

۲- اجازه دادن به تیم برای انتخاب روند (با شناخت کامل از این مسأله که تیم ضمن انتخاب، مسئول آن است که یک محصول با کیفیت را ارائه کند).

مدیر پروژه نرم افزاری که همراه با تیم مشغول به کار است باید قبل از شروع پروژه نقشها و مسئولیتها را با دقت مزید بررسی قرار دهد و تصحیح کند. تیم نیز باید مکانیسم لازم برای پاسخگویی به مسئولیتها را مشخص کند (بازبینی ها و مرورهای فنی رسمی عالی‌ترین روش تکمیل این کار می‌باشد) و هنگامی که یکی از اعضا تیم از انجام دادن این مکانیسم سرباز می‌زند، مجموعه‌ای از رهیافتهای صحیح را تعریف کند.

نقل قول

انجام دهم یا ندهیم.

کوششی در کار نیست.

بودا

(جنگ ستارگان)

هر یک از تیم های نرم افزاری با شکست های کوچکی مواجه می‌شوند. کلید اصلی برای اجتناب از جو شکست و ناکامی آن است، که فنونی براساس تیم برای فینیک و حل مشکل ایجاد شود. علاوه بر آن هر شکستی که برای هر کدام از اعضا تیم بوقوع می‌پیوندد باید به عنوان شکست تیم به آن توجه شود. و این امر به فنون جهت‌دهی شده تیمی منتهی می‌شود و به جای شانه خالی کردن از زیربار مسئولیت و بی‌اعتمادی که به سرعت در تیم‌های مسموم رشد می‌یابد به عملکردی صحیح ختم خواهد شد.

علاوه بر ۵ سهم توضیح داده شده از سوی جکمن، تیم نرم افزاری اغلب با تفاوت خصیصه‌های انسانی اعضای خود در تنازع است. بعضی از اعضا تیم درون گرا و بعضی دیگر بیرون‌گرا هستند. بعضی از اعضا اطلاعات را به صورت مستقیم جمع‌آوری می‌کنند و از حقایق غیرمتجانس و مختلف مفاهیم گسترده‌ای را استخراج می‌کنند. و سایر اعضا نیز اطلاعات را به صورت خطی پردازش می‌کنند و از روی اطلاعات فراهم شده جزئیات دقیق را گردآوری کرده و سازماندهی می‌کنند. بعضی از اعضا تیم فقط هنگامی که یک بحث منطقی و مرتب و منظم انجام می‌شود به راحتی تصمیم‌گیری می‌کنند. ولی عده دیگری از این اعضا احساساتی هستند و براساس «احساسات» خود به تصمیم‌گیری می‌پردازند.

بعضی از کارورزان تیم خواهان یک برنامه زمانی مفصل مشخص شده از سوی وظائف سازمانی هستند که آنها را در امر دستیابی به توافق و خاتمه دادن به بعضی از عناصر پروژه یاری کند. بعضی دیگر ترجیح می‌دهند که در یک محیط خودجوش تر که در آن مطالب آزاد نیز به راحتی پذیرفته می‌شوند، قرار بگیرند. بعضی از آنها سخت می‌کوشند تا چیزهایی را که قبل از یک تاریخ مشخص انجام شده است بدست آورند و بدین وسیله از استرس مربوط به رهیافت تاریخ انجام کار اجتناب می‌کنند. اما بعضی دیگر از این کارورزان به زور و در آخرین دقایق برنامه داده شده به آنها کار را انجام می‌دهند. بحث مفصل در ارتباط با روانشناسی این خصیصه‌ها و روشهایی که رئیس ماهر تیم می‌تواند بوسیله آن به کارکنان کمک کند تا با

خصیصه‌های مخالف با خصیصه‌های قبلی به کار کردن در کنار یکدیگر مشغول باشند، خارج از حیطه بحث این کتاب می‌باشد.^{۱*} با این وجود توجه به این امر که شناخت تفاوت‌های انسانی اولین گام در راه ایجاد تیم همبسته می‌باشد، مهم است.

۴-۲-۳ مسائل هماهنگی و ارتباطات

با مشکل مواجه شدن پروژه‌های نرم افزاری ممکن است دلایل و علل مختلفی داشته باشد. مقیاس^۲ بسیاری از تلاش‌های (کارهای) مربوط به توسعه، بسیار بزرگ می‌باشد که این امر موجب بوجود آمدن پیچیدگی، سردرگمی و مشکلات در هماهنگ کردن اعضای تیم می‌شود. عدم اطمینان^۳ امری معمولی و مشترک در پروژه‌های نرم افزاری می‌باشد که موجب جریان مداوم تغییراتی می‌شود که این جریان مانع تیم پروژه می‌باشد.

قابلیت استفاده داخلی^۴ به صورت یکی از مشخصه‌های کلیدی و اصلی بسیاری از سیستم‌ها درآمده است. نرم‌افزار جدید باید با نرم‌افزار موجود ارتباط برقرار کند و با محدودیت‌های از پیش تعیین شده‌ای که از سوی سیستم یا محصول تحمیل می‌شود، مطابقت داشته باشد.

این خصیصه‌های نرم افزاری مدرن، یعنی مقیاس، عدم اطمینان و قابلیت استفاده داخلی حقایق زندگی هستند. برای کنترل مؤثر این خصیصه‌ها، تیم مهندسی نرم افزاری باید شیوه‌های مناسب و مؤثری را به منظور هماهنگ کردن افرادی که کار را با یکدیگر انجام می‌دهند، ایجاد کند. برای تکمیل هماهنگی، مکانیسم‌های لازم برای روابط رسمی و غیررسمی میان اعضای تیم و روابط میان چند تیم مختلف باید مشخص شوند. روابط رسمی معمولاً از طریق نوشتن نامه ملاقات‌های رسمی و تقریباً از طریق سایر کانال‌های ارتباطی [KRA95]^۵ غیر شخصی بوجود می‌آید. روابط غیر رسمی شخصی‌تر می‌باشند. اعضای یک تیم نرم افزاری در ارتباط با یک مورد خاص دارای نظرات مشترکی هستند، هنگام بوجود آمدن مشکل از یکدیگر کمک می‌گیرند، و در برنامه‌های روزانه خود با یکدیگر در ارتباط و تماس هستند.

۱. یک مقدمه ممتاز در خصوص تیم‌های نرم افزاری را می‌توانید در [FER98] ملاحظه نمایید.

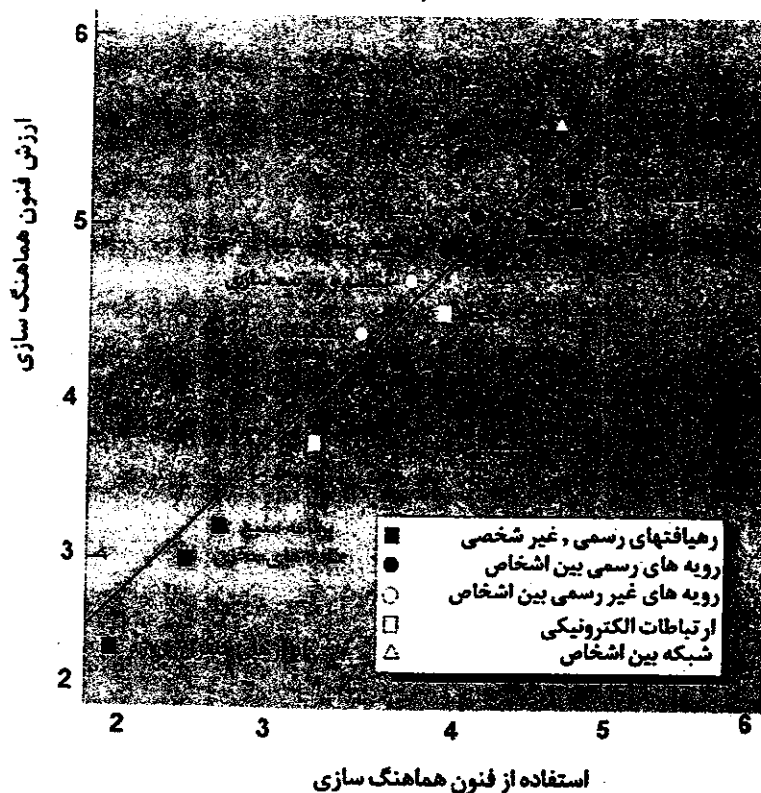
* مترجم خود در حیطه‌های مختلف خاصیت "شب امتحانی" بودن و عواقب آن را ملاحظه نموده است، نه تنها در پروژه‌ها و پاستگویی مسئولیت‌ها که در امور اقتصادی - وظائف روزمره، حتی تصمیم‌گیری‌ها و فرایندهای شخصی و غیر شخصی سیاسی و فرهنگی اینگونه‌ایم و اکثر اقدامات را در آخرین فرصت‌ها انجام می‌دهیم که عواقب آن، گاه غیر قابل جبران است.

2. scale of

3. Uncertainty

4. Interoperability

5. Kral, R. and L.



شکل ۱-۳ ارزش کاربرد فنون ارتباط و هماهنگ سازی

کرول و استریتر [KAR95] مجموعه‌ای از فنون هماهنگ کننده پروژه که به شکل زیر طبقه‌بندی

می‌شوند را مورد بررسی قرار داده‌اند:

رهیافتهای رسمی و غیر رسمی

این رهیافت شامل موارد زیر می‌باشد: اسناد و مدارک قابل تحویل مهندسی نرم افزاری (از جمله کد برنامه منبع)، نوشته‌های فنی، زمان ارزیابی پروژه، برنامه‌ها و ابزار کنترل پروژه (فصل ۷)، بوجود آمدن تغییرات مجدد و مدارک مربوطه (فصل ۱۹)، گزارشهای مربوط به پیگیری خطاها و اطلاعات سری (فصل ۳۱).

شیوه‌های رسمی و میان فردی

بر فعالیتهای تضمین کیفیت (فصل ۸) که درمورد محصول کاری مهندسی نرم افزاری بکار برده شده است، تأکید دارد. این شیوه‌ها شامل جلسات تجدید نظر رسمی، بررسی مجدد طرحها و کندها می‌باشد.

شیوه‌های غیررسمی و میان فردی

این شیوه‌ها معمولاً شامل جلسات گروهی می‌باشد که به منظور بخش اطلاعات و حل مشکلات و جمع‌آوری اطلاعات درمورد نیازهای کارمندان و توسعه انجام شده، برگزار می‌شود.



چگونه همکاری را بین
افراد تیم به وجود
آوریم؟

روابط الکترونیکی

این روابط شامل پست الکترونیکی، بردهای الکترونیکی پژوهشی و در حالت گسترده و پیشرفته شامل سیستم های گفتگو ویدئویی می باشد.

شبکه میان فردی

مذاکرات رسمی با اعضاء تیم و مذاکرات رسمی با افرادی که جزء پروژه نمی باشند اما تجارب یا اطلاعاتی دارند که می تواند به اعضاء تیم کمک کند.

برای ارزیابی میزان کارایی این تکنیکها در هماهنگ کردن پروژه، کارول و استریتر ۵۶ پروژه نرم افزاری را که شامل صدها نفر کارمند فنی بوده مورد بررسی و مطالعه قرار دادند. شکل ۱-۲ اندازه فنون هماهنگ کننده فوق الذکر و کاربرد آنها را نشان می دهد. با مراجعه به این جدول مشخص می شود که اندازه مشخص شده برای هماهنگ کنندهها و فنون مختلف روابطی در مقابل تعداد دفعات استفاده از آنها در طول یک پروژه طرح ریزی شده است. فنونی که در قسمت بالایی خط رگرسیون قرار گرفته اند، با در نظر گرفتن تعداد دفعات آنها به عنوان فنون با ارزش و مفید در نظر گرفته می شوند. اما فنونی که در پایین خط رگرسیون واقع شده اند از ارزش کثیری برخوردار هستند. جالب توجه است که بدانید شبکه ارتباطی بین فردی به عنوان تکنیکی که دارای بالاترین میزان هماهنگی و بالاترین اندازه روابط است، [KRA95] مشخص شد. همچنین ذکر این نکته نیز حائز اهمیت است که مکانیسم های اولیه تضمین کیفیت نرم افزاری (بازبینی نیازمندیها و طراحی) نسبت به ارزیابی های بعدی کد برنامه منبع (بازرسی های مربوط به کدها) از ارزش بیشتری برخوردار هستند.

۳-۳ محصول

مدیر پروژه نرم افزاری ممکن است از همان ابتدای شروع پروژه مهندسی نرم افزاری با مشکل مواجه شود. در این پروژه برآوردهای کمی و طرح های سازمان یافته مورد نیاز می باشند. اما اطلاعات خالص و قابل اطمینان در دسترس نمی باشند. تجزیه و تحلیل مفصل در ارتباط با نیازمندیهای نرم افزاری می تواند اطلاعات لازم برای برآوردهای کمی را فراهم کند. اما تکمیل این تجزیه و تحلیل ها اغلب هفته ها و ماهها به طول می انجامد. و بدتر آنکه، ممکن است نیازمندیها قابل تغییر باشند و در روند اجرای پروژه به صورت منظم و مداوم تغییر یابند. اما حالا در این شرایط یک طرح مورد نیاز می باشد.

بنابراین ما باید محصول و مسئله ای را که قرار است حل نماییم در همان ابتدای شروع پروژه، مورد بررسی قرار دهیم، حداقل باید - - - - - طه و دامنه محصول مشخص و تعیین شود.

۱-۳-۳ دامنه نرم افزار

اولین فعالیت مدیریت پروژه، نرم افزاری تعیین دامنه نرم افزار^۱ می باشد. دامنه، با پاسخ دادن به

سؤالات زیر مشخص می شود:

بافت^۲: چگونه می توان نرم افزاری ساخت که با سیستمی یا محصولی یا بافت تجاری بزرگتر هماهنگ

باشد و محدودیتها و قیود ناشی از این امر چه چیزهایی می باشند؟

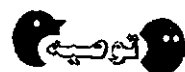
اشیاء اطلاعاتی^۳: کدام اشیاء داده ای است که مشتری به گونه ای واضح و مرئی آنها را می بیند و

درک می کند و (فصل ۱۱) به عنوان خروجی اصلی حاصل از نرم افزار بوجود می آورند؟ برای ورودی چه

اشیاء داده ای مورد نیاز می باشد؟

کارکرد و عملکرد^۴: نرم افزار برای تبدیل داده های ورودی به داده های خروجی چه عملی را انجام

می دهد. آیا ویژگی های عملکردی خاصی باید مورد بررسی قرار گیرد؟



اگر ویژگیهای نرم

افزار در دست ساخت

را نمی توانید محدود

کنید، تعیین این

ویژگیها را به عنوان

یک پروژه مهم ریسک

در نظر گیرید.

دامنه پروژه نرم افزاری باید در سطوح مدیریتی و فنی غیر مبهم قابل فهم و واضح و مشخص باشند.

صورت وضعیت دامنه نرم افزاری باید محدود و مشخص باشد. یعنی داده های کمی (به عنوان مثال

تعداد کاربران همزمان، اندازه لیست پستی، ماکزیمم زمان مجاز برای پاسخ) باید به صورت کاملاً مشخص و

صریح بیان شده باشد با محدودیتها یا حدودها (به عنوان مثال هزینه محصول محدود کننده اندازه حافظه

است) باید مورد توجه قرار گیرند و فاکتورهای تخفیف دهنده (به عنوان مثال الگوریتم مطلوب به خوبی

درک می شود و در ++C وجود دارد) باید توضیح داده شوند.

۲-۳-۳ تجزیه و شکستن مسئله

تجزیه مشکل^۵ که در بعضی مواقع تفکیک سازی مشکل^۶ یا تشریح مشکل^۷ نامیده می شود، فعالیتی

است که در مرکز تجزیه و تحلیل نیازمندیهای نرم افزاری قرار دارد (فصل ۱۱). در طول انجام فعالیت

مربوط به مشخص کردن دامنه، هیچ تلاشی برای تجزیه کامل مشکل انجام نمی شود. بلکه تجزیه در دو

محدوده اصلی و عمده ذیل بکار برده می شود: ۱- عملکردهایی که باید ارائه شوند. ۲- روند استفاده شده

برای ارائه این عملکردها.



برای ساخت یک طرح

پروژه قابل دفاع اولویت

با شکستن مسئله برای

حل آن می باشد

1. software scope

2. Content

3. Information Objectives

4. Function and Performance

5. Problem Decomposition

6. partitioning

7. elaboration

انسانها در برخورد با مشکلات پیچیده مایلند از راهبرد تجزیه و غلبه استفاده کنند. به زبان ساده تر ابتدا یک مشکل پیچیده را به مشکلات کوچکتری که قابلیت کنترل و مهار آنها بیشتر می باشد، تقسیم می کنند. این راهبرد راهبردی است که همزمان با آغاز برنامه ریزی پروژه بکار برده می شود. عملکردهای نرم افزاری توضیح داده شده در قسمت دامنه نرم افزار مورد ارزیابی و اصلاح قرار می گیرند تا قبل از آغاز برآوردها اطلاعات بیشتری فراهم کنند. از آنجایی که برآوردهای هزینه و برنامه به صورت عملی جهت دهی می شوند. بنابراین تجزیه مشکل تا حد مشخص سودمند خواهد بود.

به عنوان مثال پروژه های را در نظر بگیرید که محصول ویژه پردازی را بوجود می آورد. مشخصه های منحصر به فرد این محصول عبارتند از: صدای مستمر ورودی صفحه کلید، مشخصه های بسیار پیچیده ویرایش خودکار، قابلیت صفحه آرای، شاخص گذاری خودکار و فهرست مطالب و غیره. مدیر طرح ابتدا باید صورت وضعیت دامنه را که محدود کننده این ویژگی ها (و همچنین محدود کننده سایر عملکردهای عملی تر همچون ویرایش، کنترل فایل، تولید سند و غیره) است، ایجاد نماید. به عنوان مثال آیا ورودی صدای مستمر به این امر نیاز خواهد داشت که محصول بوسیله کاربر آماده شود؟ ویژگی ویرایش کپی خودکار چه قابلیت هایی را فراهم خواهد کرد؟ و میزان پیچیدگی قابلیت صفحه آرای چه قدر خواهد بود.

با مشخص شدن صورت وضعیت دامنه، اولین مرحله تفکیک سازی به صورت طبیعی اتفاق می افتد. تیم پروژه متوجه می شوند که بخش بازاریابی با مشتریان بالقوه ای در مورد این پروژه به بحث و گفتگو پرداخته است و همچنین متوجه می شوند که عملکردهای ذیل باید به عنوان بخشی از ویرایش کپی خودکار انجام شوند: ۱- کنترل دیکته لغات. ۲- کنترل گرامر جملات. ۳- کنترل منابع و مأخذ برای اسناد بزرگ (به عنوان مثال آیا مرجع موجود در ورودی فهرست در لیست ورودی های موجود در فهرست وجود دارد یا خیر؟). ۴- کنترل اعتبار مرجع فصلها و بخشهای مختلف در اسناد بزرگ. هر یک از این ویژگی ها نشانگر عملکردی فرعی می باشند که باید در نرم افزار اجرا شوند و در مقابل در صورتیکه تجزیه و تفکیک بتواند موجب ساده تر شدن طرح ریزی شود، می توان در هر یک از این ویژگی ها اصلاحات بیشتری انجام داد.

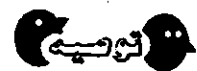
۴-۳ فرآیند

مراحل کلی مشخص کننده فرآیند یک نرم افزار - یعنی تعریف، توسعه و پشتیبانی - در مورد تمام نرم افزارها قابل اجرا خواهند بود. مشکل موجود در این مرحله انتخاب مدل فرآیند متناسب با نرم افزار می باشد که قرار است توسط تیم پروژه طرح ریزی شود. در فصل دوم مجموعه گسترده ای از نمونه های طراحی نرم افزاری مورد بحث و بررسی قرار گرفت که به ترتیب عبارتند از:

- مدل ترتیبی خطی.
- مدل نمونه سازی.
- مدل RAD



یک تکنیک مفید برای تجزیه و شکستن مسئله "تجزیه گرامری" نام دارد، که در فصل ۱۲ ارائه شده است.



هنگامیکه مدل فرآیند انتخاب می شود، با حداقل مجموعه از وظائف کاری و محصولات کاری توصیف می گردد. به این طریق محصولی با کیفیت بالا داشته، از نیروی کاری زیاده از حد نیز اجتناب خواهیم نمود.

- مدل افزایشی.
- مدل حازونی.
- مدل حازونی WIN-WIN.
- مدل توسعه مبتنی بر اجزاء.
- مدل توسعه همروند.
- مدل شیوه های رسمی.
- مدل فنون نسل چهارم.

مدیر پروژه باید تصمیم گیری کند که کدام یک از نمونه های فرآیندهای موارد زیر مناسب می باشند

۱- مشتریانی که این محصول را سفارش داده اند و هر دمی که کار مربوط به آن را انجام

خواهند داد.

۲- مشخصه های خود محصول.

۳- محیط پروژه که در آن تیم نرم افزاری مشغول به کار است.

پس از انتخاب مدل پردازشی، تیم، طرح اولیه پروژه را براساس مجموعه ای از فعالیتهای مشترک و رایج چهارچوب فرآیند تعریف خواهد کرد. پس از مشخص شدن طرح ابتدایی، تفکیک فرآیند آغاز می شود. یعنی یک طرح کامل که منعکس کننده وظایف کاری مورد نیاز برای پر کردن چارچوب فعالیتها است باید در این مرحله ایجاد شود. این فعالیتها به طور خلاصه در فصل بعدی گنجانده شده است و بحث مفصل در ارتباط با آن در فصل ۷ آورده شده است.

۱-۴-۲ تلفیق^۱ محصول و فرآیند

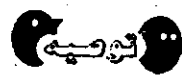
برنامه ریزی پروژه با تعیین محصول و فرآیند آغاز می شود. هر عملکردی که قرار است بوسیله تیم نرم افزاری طراحی شود باید از مجموعه چارچوب فعالیتهایی که برای سازمان نرم افزاری تعریف شده اند، عبور کند. فرض کنید که این سازمان مجموعه چارچوب فعالیتهای زیر را برگزیده باشد (فصل ۲):

- ارتباط با مشتری^۲: کارهایی که برای کشف مؤثر نیازمندیها میان توسعه دهنده و مشتری

لازم می باشند.

- طرح ریزی^۳: کارهایی که برای تعریف منابع، خطوط زمانی و سایر اطلاعات مربوط به پروژه

نیاز می باشد.



به خاطر داشته باشید

که فعالیتهای

چارچوبی برای سیر هر

پروژه ای کاربرد دارد-

استثنائی وجود

نخواهد داشت.

1. Melding

2. Customer communication

3. Planning

- | مهندسی | تحلیل ریسک | طرح ریزی | ارتباط با مشتری | فعالیت‌های چارچوب مشتری فرآیند |
|--------|------------|----------|-----------------|--------------------------------|
| | | | | فعالیت‌های مهندسی نرم افزار |
| | | | | کارکردهای محصول |
| | | | | ورودی متن |
| | | | | ویرایش و قالب بندی |
| | | | | اصلاح خودکار |
| | | | | آرایش صفحه و صفحه بندی |
| | | | | شاخص بندی خود کار و TOC |
| | | | | مدیریت فایل |
| | | | | تولید مستندات |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

نقد

1. Risk analysis
2. Engineering
3. Construction and release
4. Customer evaluation

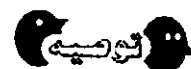
ارقام نشاندهنده عملکرد می‌باشند) در ستون سمت چپ لیست شده‌اند. چارچوب فعالیت‌ها در ردیف بالا ذکر شده‌اند.^۱ کارهای مربوط به طراحی نرم‌افزار (برای هر چارچوب فعالیتی) در ردیف پایین قرار خواهند گرفت. کار مدیر پروژه (و سایر اعضای تیم) برآورد منابع مورد نیاز هر یک از ردیف‌های موجود در این شاخص (نمودار) می‌باشد؛ آنها همچنین باید تاریخ شروع و اتمام هر یک از کارها را در ستون مربوطه ثبت کنند. همچنین تاریخ شروع و اتمام محصولات کاری که در نتیجه انجام هر یک از کارهای موجود در ستونهای این نمودار بوجود می‌آیند را ثبت کنند. این فعالیتها در فصلهای ۵ و ۷ مورد بررسی قرار گرفته‌اند.

۲-۴-۳ شکستن فرآیند^۲

تیم نرم‌افزاری باید در انتخاب الگوی مهندسی نرم‌افزار بسیار انعطاف پذیر باشد. این پارادایم باید بهترین الگو طراحی برای پروژه و کارهای طراحی نرم‌افزاری که مدل فرآیند را پس از انتخاب تعیین می‌کنند، باشد. پروژه‌های نسبتاً کوچک که مشابه کارهای قبلی می‌باشد با استفاده از رهیافت ترتیبی خطی به بهترین نحو تکمیل می‌شوند. در صورتیکه محدودیت زمانی بسیار شدیدی برای انجام پروژه وجود داشته باشد و مشکل را بتوان به سختی به بخشهای مجزا تفکیک نمود، مدل RAD شاید بهترین انتخاب باشد. اگر موعد تحویل به قدری محدود و فشرده باشد که انجام عملکرد کامل آن به صورت منطقی ممکن نباشد، راهبرد افزایشی بهترین راه حل خواهد بود. همچنین پروژه‌هایی با سایر ویژگی‌ها (به عنوان مثال نیازمندیهای نامشخص، تحول شگرف در فناوری، مشتریان سرسخت، توانایی استفاده مجدد) به انتخاب سایر مدل‌های پردازشی منتهی خواهد شد.^۲

با انتخاب مدل فرآیند، چارچوب روند معمول (CPF)^۳ نیز با آن تطابق پیدا می‌کند. در هر موردی CPF فوق‌الذکر - یعنی ارتباط با مشتری، برنامه‌ریزی، تجزیه و تحلیل احتمال خطر، طراحی، ساخت، انتشار، ارزیابی مشتری - را می‌توان متناسب با نمونه انتخاب کرد. این مسأله درمورد مدل‌های خطی، مدل‌های افزایشی یا تکراری، مدل‌های تکامل یافته و حتی درمورد مدل‌های سوارکردنی همزمان نیز صادق است. CPF ثابت و غیرمتغیر است و به عنوان پایه‌ای برای تمام کارهای نرم‌افزاری که توسط سازمان نرم‌افزاری اجرا می‌شوند، عمل می‌کند.

اما کار واقعی رسیدگی محاسباتی دارای تفاوت‌هایی می‌باشد. تجزیه پردازشی هنگامی آغاز می‌شود که مدیر پروژه این سؤال را مطرح می‌سازد که: «چگونه می‌توان این فعالیت CPF را تکمیل نمود؟» به عنوان



همواره چارچوب مشترک فرآیند (CPF) را بکار ببرید. صرف‌نظر از اندازه، کاردینالیتی یا نوع پروژه، وظائف کاری محتمل است که متفاوت باشند. ولی CPF انیستگونه نخواهد بود.

۱. باید متذکر شد که وظائف کاری، با نیازهای خاص پروژه‌های نرم‌افزاری مطابقت داده خواهد شد. این عنوان در فصل ۷ و سایت وب SEPA توضیح داده شده است.

2.Process Decomposition

۳. به خاطر آورد که ویژگیهای پروژه شدیداً بر ساختار تیمی که انجام کار را عهده دار است، تاثیر گذار خواهد بود. بخش

4.Common Process Framework

مثال یک پروژه کوچک و نسبتاً ساده برای فعالیت ارتباط با مشتری به کارهای محاسباتی زیر نیاز خواهد داشت:

- ۱- فراهم کردن لیستی از مسائل رفع کننده ابهام.
 - ۲- ملاقات با مشتری برای مورد بررسی قرار دادن مسائل رفع کننده ابهام.
 - ۳- گسترش مشترک تعریف دامنه.
 - ۴- بررسی تعریف دامنه با در نظر گرفتن تمام مشکلات.
 - ۵- تغییر تعریف دامنه در صورت نیاز.
- این اتفاق ها ممکن است در دوره ای که طول آن کمتر از ۴۸ ساعت است اتفاق بیافتد. این مسائل تجزیه پردازشی متناسب با پروژه کوچک و نسبتاً ساده را نشان می دهند.
- حالا، یک پروژه پیچیده تر که دارای دامنه گسترده تر و اثر کاری مهم تر است در نظر می گیریم. چنین پروژه ای برای فعالیت ارتباط با مشتری^۱ به کارهای محاسباتی زیر نیازمندیهای خواهد داشت:



مثال دوم: مدل تطبیقی

- ۱- بررسی تقاضای مشتری.
- ۲- طرح ریزی و برنامه ریزی برای انجام ملاقات رسمی و آسان با مشتری.
- ۳- انجام تحقیق و پژوهش برای تعریف راه حلهای پیشنهادی و رهیافتهای موجود.
- ۴- آماده کردن اسناد کاری و آماده کردن یک دستور جلسه برای جلسات رسمی.
- ۵- تنظیم و اداره کردن جلسات.
- ۶- گسترش مشترک مشخصات جزئی که داده ها، عملکرد و ویژگی های رفتاری نرم افزار را منعکس می کنند.

۷- بررسی صحت هر یک از مشخصات جزئی، و بررسی عدم تناقض و عدم ابهام این مشخصات.

۸- گردآوری مشخصات جزئی در اسناد مربوطه.

۹- بررسی اسناد مربوطه با تمام مشکلات و مسائل مربوط به آنها.

۱۰- تغییر اسناد مربوطه در صورت نیاز.

در این دو پروژه چارچوب فعالیتی که ما آن را «ارتباط با مشتری» می نامیم، اجرا می شود، اما در تیم پروژه اول (یعنی پروژه ساده و نسبتاً کوچک) نیمی از کارهای محاسباتی مهندسی نرم افزاری انجام شده در پروژه دوم (پروژه پیچیده تر) اجرا می شود.

۳-۵ پروژه

به منظور دستیابی به یک پروژه نرم افزاری موفق، ابتدا باید به این مطلب پی برد که چه مشکلی وجود دارد (طوری که بتوان از مشکلات اجتناب نمود) و چگونه می توان آن مشکل را برطرف نمود. در یک

نقل قول

از هر علامت

تشکلیت پروژه

سیستمهای

اطلاعاتی حداقل ۷

نای آنها پیش از

رسیدن به برنامه

نویسی مشهور

می باشد:

جان ریل

مقاله بسیار عالی درمورد پروژه‌های نرم‌افزاری، جان ریل [REE99]^۲ ده علامت (نشانه) را معرفی می‌کند که این علائم نشان می‌دهند که پروژه سیستم‌های اطلاعاتی در معرض خطر قرار گرفته‌اند. این ۱۰ نشانه (علامت) عبارتند از:

- ۱- متخصصان نرم‌افزاری نیازهای مشتری خود را درک نمی‌کنند.
- ۲- حوزه عملکرد محصول ضعیف تعریف شده است.
- ۳- تغییرات به خوبی کنترل نمی‌شوند.
- ۴- فناوری انتخاب شده تغییر می‌یابد.
- ۵- نیازهای تجاری در حال تغییر هستند.
- ۶- موعدهای مقرر و مهلت‌ها غیرواقعی هستند.
- ۷- کاربران مقاوم هستند.
- ۸- ضمانت این سیستم‌ها از بین رفته است [یا این ضمانت هرگز به صورت مناسب و درست حاصل شده است].

۹- افراد متخصص با مهارت‌های مناسب در تیم پروژه وجود ندارد.

۱۰- مدیران و (کارورزان) از بکار بردن بهترین تمرین‌ها و درس‌های فراگرفته شده اجتناب می‌کنند.

متخصصان نامی صنایع هنگام بحث درمورد پروژه‌های سخت نرم‌افزاری اغلب به قانون ۹۰-۹۰-۱۰ مراجعه می‌کنند. اولین ۹۰ درصد یک سیستم ۹۰ درصد تلاش و زمان اختصاص یافته را جذب می‌کند و ۱۰ درصد آخر آن نیز ۹۰ درصد تلاش و زمان اختصاص یافته را جذب می‌کند [ZAH94]^۳. مسائلی که به استفاده از قانون ۹۰-۹۰-۱۰ منتهی می‌شوند در ۱۰ علامت (نشانه) فوق‌الذکر وجود دارند. اما این قانون دارای جنبه‌های منفی نیز می‌باشد! چگونه یک مدیر برای اجتناب از مشکلات فوق‌الذکر می‌تواند عمل کند؟ ریل [REE99]^۴ رهیافت پنج قسمتی قضاوت صحیح را درمورد پروژه‌های نرم‌افزاری پیشنهاد می‌دهد:

۱- کار را به روشی صحیح آغاز کنید. این امر با تلاش زیاد برای درک مشکلی که باید حل شود و سپس با تنظیم توقعات و انتظارات واقعی هر شخصی که به نوعی در این پروژه دخیل می‌باشد، تکمیل خواهد شد. این کار با بوجود آوردن تیم مناسب و دادن استقلال داخلی، قدرت و فناوری موردنیاز کار به این تیم، تقویت خواهد شد.

۲- نیروی سوق‌دهنده این کار را حفظ کنید. بسیاری از پروژه‌ها آغاز خوبی دارند. اما به تدریج از هم پاشیده می‌شوند. برای حفظ نیروی سوق‌دهنده پروژه، مدیر پروژه باید انگیزه‌هایی را برای حفظ



ارجاع به وب

آرایه گسترده‌ای از منابع که هم مدیران مجرب پروژه‌ها را یاریگر خواهد بود و هم تازه‌کارا، در آدرس‌های زیر یافت می‌شود:

www.pmi.org
www.4pm.com
www.projectmanagement.com

1. customer communication activity

2. Reel, J.S.

3. Zahniser, R.

4. Reel, J.S.

عملکرد پرسنل در حداقل مقدار این عملکرد، ایجاد کند و تیم در هر کاری باید کیفیت عملکرد خود تأکید داشته باشد و مدیر ارشد نیز باید هر کار ممکن را انجام دهد تا تحت تأثیر روش و مسیر تیم قرار نگیرد.

۳- پیشرفت حاصله را دنبال کنید. درمورد پروژه های نرم افزاری، باوجود آمدن محصولات کاری (به عنوان مثال، مشخصات، کد برنامه منبع، مجموعه هایی از موارد آزمونی) و با به تصویب رسیدن آنها (استفاده از بازبینی های فنی و رسمی) به عنوان بخشی از فعالیت تضمین کیفیت، پیشرفت کاری نیز دنبال می شود. علاوه بر آن فرآیند نرم افزاری و سنجش های پروژه ای (فصل ۴) را می توان جمع آوری نمود و برای ارزیابی

پیشرفت در مقابل مقدار متوسط توسعه و گسترش سازمان از نقطه نظر توسعه نرم افزاری آن بکار برد.

۴- در پروژه خود تصمیم گیری صحیح و عاقلانه انجام دهید. درصورت لزوم تصمیم گیریهای مدیر پروژه و تیم نرم افزاری باید «به صورت ساده باشد» و درصورت امکان تصمیم بگیرید که از نرم افزارهای تجاری حاضر و آماده و یا عناصر نرم افزاری موجود استفاده کنید؛ تصمیم بگیرید که هنگام وجود رهیافتهای استاندارد، از رابطهای سفارشی عادی اجتناب کنید. تصمیم بگیرید که خطرات احتمالی واضح و مشخص را شناسایی و از آنها اجتناب کنید. تصمیم بگیرید که برای کارهای پرخطر و پیچیده بیشتر از مدت زمانی که شما فکر می کنید برای آن کار مورد نیاز است به آن اختصاص دهید (شما به هر دقیقه نیز نیاز خواهید داشت).

۵- یک تجزیه و تحلیل نیز پس از اتمام پروژه انجام دهید. یک مکانیسم ثابت برای استخراج و خلاصه کردن درسهای فراگرفته شده از هر پروژه ایجاد نمایید. برنامه های طرح ریزی شده و واقعی را ارزیابی کنید. استانداردهای پروژه نرم افزاری را گردآوری نموده و مورد تجزیه و تحلیل قرار دهید، فیدبک لازم را از اعضا تیم و مشتریان بدست آورید و یافته ها را به شکل کتبی ثبت کنید.

۳-۶ اصل W⁵HH

در یک مقاله عالی در ارتباط با پروژه ها و فرآیند های نرم افزاری، باری بوهم [BOE96]^۱ اظهار

داشت:

«شما به یک اصل سازمان یافته نیاز دارید که برای فراهم کردن طرحهای ساده برای پروژه های ساده با کاهش تدریجی مواجه شود.» بوهم رهیافتی را پیشنهاد می کند که این رهیافت اهداف، برنامه ها و نقاط عطف پروژه، مسئولیتها، رهیافت مدیریتی و فنی و منابع مورد نیاز برای پروژه را مورد بررسی قرار می



برای توسعه طرح یک
پروژه، چه سوالاتی
باید به ترتیب پرسیده
شود؟

دهد. وی بعد از مطرح ساختن مجموعه‌ای از سوالات که به تعریف مشخصه‌های کلیدی پروژه و طرح حاصله پروژه منتهی شد، این اصل را، اصل WWWWHH^۱ نامید. این مجموعه سوالات عبارتند از:

چرا سیستم توسعه می‌یابد؟ پاسخ به این سؤال تمام افراد را قادر می‌سازد تا میزان اعتبار دلائل کار نرم‌افزاری را ارزیابی کنند. به عبارت دیگر آیا هدف کاری می‌تواند توجه‌کننده زمان و پول صرف شده باشد.

چه کاری تا چه زمانی انجام خواهد شد؟ پاسخ این سؤال به تیم کمک خواهد کرد تا بوسیله تشخیص کارهای کلیدی پروژه و نقاط عطف موردنیاز مشتری یک برنامه پروژه‌ای ایجاد نمایند.

چه کسی مسئول یک عملکرد است؟ در اوایل این فصل ما به این نکته اشاره کردیم که نقش و مسئولیت هر یک از اعضای تیم باید تعریف شده باشد. پاسخ به این سؤال می‌تواند موجب تکمیل این تعریف شود.

این مسئولیت‌ها به صورت سازمانی در کجا واقع شده‌اند؟ تمام نقشها و مسئولیتها فقط در درون خود تیم نرم‌افزاری واقع نشده‌اند بلکه مشتریان، کاربران و سهامداران نیز دارای مسئولیت‌هایی می‌باشند.

چگونه یک کار به صورت فنی و مهارتی انجام خواهد شد؟ با مشخص شدن دامنه محصول، راهبرد مدیریت و فنی لازم برای پروژه نیز باید تعریف شود.

چه میزان از هر یک از منابع موردنیاز می‌باشد؟ پاسخ به این سؤال با گسترش دادن و توسعه پاسخ‌های داده شده به سوالات قبلی بدست می‌آید.

اصل W⁵HH^۲ بوهم صرف نظر از اندازه و پیچیدگی پروژه نرم‌افزاری قابل استفاده می‌باشد. سوالات فوق‌الذکر می‌توانند طرحی عالی از برنامه ریزی را برای مدیران پروژه و تیم نرم‌افزاری فراهم کنند.



طرح پروژه نرم‌افزاری

۳-۷ اقدامات بحرانی^۲

شورای Air Lie^۳ لیستی از کارهای حساس و مهم نرم‌افزاری را در ارتباط با مدیریت مبتنی بر عملکرد فراهم کرده‌اند. این کارها به صورت ثابت و پیوسته از سوی پروژه‌ها و سازمانهای موفق نرم‌افزاری که عملکرد خط انتهای آنها دائماً بیش از عملکرد [AIR99]^۴ متوسط صنایع می‌باشد، مورد استفاده قرار

1. Why – What – When – Who – Where – How – How much

2. Critical Practices

۳. هیئت "ایر لای" یک تیم خبره مهندسی نرم‌افزار است، که دیپارتمان دفاعی ایالات متحده برای رهنمود در خصوص بهترین عملکردهای مدیریت پروژه نرم‌افزاری و مهندسی نرم‌افزار تشکیل داده است.

4. Airline Council

گرفته و به عنوان کارهای حساس در نظر گرفته می شوند. این شورا در تلاش برای کمک کردن به یک سازمان نرم افزاری در تعیین این امر که آیا یک پروژه خاص کارهای حساسی را اجرا کرده است یا خیر، مجموعه ای از سوالات [AIR99] «دقیق و حساس» را برای یک پروژه گسترش داده اند.^۱

مدیریت رسمی ریسک و خطر. ۱۰ خطر عمده ای که متوجه این پروژه هستند چه چیزهایی می باشند؟ در مورد هر کدام از احتمالات خطر چه چیزی موجب می شود که این احتمال خطر تبدیل به مشکل شود. و در صورت تبدیل احتمال خطر به مشکل، چه تأثیراتی بر پروژه خواهد آمد؟

هزینه عملی و برآورد برنامه اندازه فعلی تخمین زده شده برای کاربرد نرم افزار که در عملکرد از آن استفاده خواهد شد، چه قدر می باشد؟ و این اندازه چگونه مشخص شده است؟

مدیریت پروژه مبتنی بر متریک ها. آیا شما برنامه هایی دارید که علائم اولیه بروز مشکل را به خوبی و به موقع نشان دهد، در صورت وجود چنین برنامه هایی، بی ثباتی شرایط فعلی آنها چه چیزهایی می باشد.

ردگیری مقادیر بدست آمده. آیا شما اندازه های استاندارد بدست آمده را به صورت ماهانه گزارش می کنید؟ اگر چنین کاری را انجام می دهید، آیا این استانداردها از شبکه فعالیتی کارا برای تلاش کنونی انجام شده و برای تحویل بعدی، محاسبه شده است؟

پیگیری معایب در مقابل اهداف کیفیتی: آیا شما معایب مشخص شده در هر بازرسی (بازبینی) را دنبال می کنند و تعداد این معایب را به صورت دوره ای گزارش می کنید. آیا تعداد معایب مشخص شده بوسیله آزمون اجرایی انجام شده از آغاز برنامه و تعداد معایبی که در حال حاضر ایجاد می شوند و یا از بین می روند را به صورت دوره ای گزارش می کنید؟

مدیریت برنامه هایی که کارکنان نسبت به آن آگاهی و بینش دارند. عملکرد هر یک از کارکنان در سه ماه گذشته در برابر هر یک از تهیه کنندگان و توسعه دهندگان دخیل در توسعه نرم افزار برای این سیستم چگونه بوده است؟

اگر تیم پروژه نرم افزاری نتواند به این سوالات پاسخ دهند یا پاسخ آنها به این سوالات به صورت ناقص باشد، بررسی کامل و جامع کارهای پروژه ضروری خواهد بود. هر یک از کارهای حساس فوق الذکر در بخش دوم این کتاب به صورت مفصل مورد بررسی قرار خواهد گرفت.



نگاه سریع به پروژه

Airlie

۱. تنها فعالیتهای بحرانی "جامعیت پروژه" در اینجا مورد اشاره واقع شده است. دیگر موارد ممتاز در فصلهای آینده توضیح داده خواهند شد.

۳-۸ خلاصه

مدیریت پروژه نرم‌افزار یک فعالیت پوشش دهنده (چتری) در محدوده مهندسی نرم‌افزار می‌باشد و این مدیریت قبل از آغاز هر فعالیت فنی آغاز می‌شود و در طول تعریف، توسعه و حمایت نرم‌افزار کامپیوتر ادامه می‌یابد.

چهار مورد بر مدیریت پروژه نرم‌افزاری تأثیر بسزایی دارند. این چهار مورد عبارتند از کارکنان، محصول، فرآیند و پروژه. کارکنان باید در تیم‌های مناسب و مؤثری سازمان بندی شوند و برای انجام کار نرم‌افزاری با کیفیت بالا تشویق و ترغیب شوند و باید با یکدیگر هماهنگ شده و همکاری کنند تا به سطح مؤثر و سودمند ارتباطی برسند و ملزومات و نیازمندیهای محصول باید از مشتری تا توسعه دهنده مشخص شود و به اجزاء سازنده آن تقسیم (تفکیک) شود و برای استفاده و کار کردن تیم نرم‌افزاری آماده شود. فرآیند باید با کارکنان و مشکل موجود هماهنگ باشد. یک چارچوب فرآیند مشترک انتخاب می‌شود و نمونه مناسب مهندسی نرم‌افزار بکار برده می‌شود و مجموعه‌ای از کارهای محاسباتی نیز برای انجام شدن کار انتخاب می‌شود. و در نهایت پروژه باید به طریقی سازماندهی شود که موجب موفقیت تیم نرم‌افزاری شود. عنصر اصلی و محوری هر پروژه نرم‌افزاری کارکنان آن پروژه می‌باشند. می‌توان مهندسان نرم‌افزار را در تعدادی از ساختارهای مختلف تیمی که از سلسله مراتب کنترل سنتی تا نیم‌های «نمونه آزاد» متفاوت می‌باشند، تقسیم بندی نمود.

می‌توان فنون متنوع ارتباطی و هماهنگی را برای حمایت از کار تیم بکار برد. به طور کلی پژوهشهای رسمی و ارتباط رودررو (فرد با فرد) برای شرکت کنندگان در این پروژه ارزش بسیاری دارد. فعالیت مدیریت پروژه شامل اندازه‌گیریها و استانداردها، برآورد، تجزیه و تحلیل احتمال خطر، زمان‌بندی، پیگیری و کنترل می‌باشد. هر یک از این موضوعات در فصلهای بعدی مورد بررسی قرار خواهند گرفت.

مسایل و نکاتی برای تفکر و تعمق بیشتر

۱-۳ براساس اطلاعات ارائه شده در این فصل و براساس تجارب خود یک مجموعه «ده فرمان» را برای قدرتمند کردن مهندسان نرم افزار توسعه دهید. یعنی ۱۰ دستورالعمل بنویسید که به متخصصان نرم افزاری کمک کند تا بتوان کامل خود به کار ادامه دهند.

۲-۳ مدل کامل قابلیت مدیریتی کارکنان مؤسسه مهندسی نرم افزاری (PM-CMM) به «تواهی کلیدی کار» که موجب پرورش متخصصان نرم افزاری خوب و ماهر می شود، با دقت توجه می کند. مدیر شما تجزیه و تحلیل کردن و خلاصه کردن یک KPA را به شما محول می کند.

۳-۳ سه موقعیت واقعی زندگی را توضیح دهید که در آن مشتری و کاربر نهایی یک نفر می باشند. سه موقعیت را توصیف کنید که مشتری و کاربر نهایی دو فرد متفاوت هستند.

۴-۳ تصمیم گیریهای انجام شده از سوی مدیریت ارشد می تواند تأثیر قابل ملاحظه ای بر کارایی تیم مهندسی نرم افزاری داشته باشد. پنج مثال (مورد) بیان کنید که نشان دهنده صحت این مطلب باشند.

۵-۳ یک نسخه از کتاب وینبرگ را مرور کنید و خلاصه ای از مطالبی که باید در هنگام بکار بردن مدل MOI در نظر گرفته شوند را در دو یا سه صفحه یادداشت کنید.

۶-۳ شما به عنوان مدیر پروژه ای در یک سازمان سیستم های اطلاعاتی برگزیده شده اند. کار شما این است که کاربردی ایجاد کنید که مشابه با کاربرد تیم باشد، هرچند کاربردی که شما بوجود می آورید بزرگتر و پیچیده تر از کاربردی است که از سوی اعضای تیم ایجاد شده است. شرایط به طور کامل از سوی مشتری ثبت شده است. شما چه نوع ساختار تیمی را انتخاب می کنید و به چه دلیل این ساختار تیمی را برمی گزینید؟ چه نوع مدل فرایند نرم افزاری را انتخاب می کنید و به چه دلیل؟

۷-۳ شما در یک شرکت کوچک محصولات نرم افزاری به عنوان مدیر پروژه منصوب شده اید. کار شما در این شرکت این است که یک محصول پیشرفته بوجود آورید سخت افزار واقعی نهایی را با پیشرفته ترین نرم افزار ترکیب کند. از آنجایی که رقابت در بازارهای داخلی محصولات شدید است، بنابراین فشار قابل ملاحظه ای برای انجام کار و تحویل آن وجود دارد. کدام یک از ساختارهای تیمی را انتخاب خواهید کرد و به چه دلیل؟ کدام یک از مدل های فرایند نرم افزاری را انتخاب خواهید کرد و به چه دلیل؟

۸-۳ شما به عنوان مدیر پروژه ای یک شرکت بزرگ محصولات نرم افزاری منصوب شده اید. وظیفه شما در این شرکت نظارت بر توسعه نسخه نسل جدی نرم افزار ویژه پردازی که به صورت گسترده مورد استفاده قرار می گیرد، می باشد. از آنجایی که رقابت شدید است، موعد تحویل فشرده و کوتاه مدتی مشخص و تعیین می شود. کدام یک از ساختارهای تیمی را در این شرایط انتخاب می کنید و به چه دلیل؟ کدام یک از مدل های فرایند نرم افزاری را در این شرایط انتخاب می کنید و به چه دلیل؟

۹-۳ شما به عنوان مدیر پروژه نرم افزاری شرکتی انتخاب شده اید که خدمات خود را به دنیای مهندسی ژنتیک ارائه می دهد. وظیفه شما در این شرکت نظارت بر توسعه محصول نرم افزاری جدیدی

است که خط سیر تقسیمات ژنتیکی را تسریع می‌کند. در اینجا کار به سرعت تحقیق و توسعه می‌باشد، اما هدف تولید یک فرآورده نرم افزاری برای سال آینده می‌باشد. در این شرایط شما کدام یک از ساختارهای تیمی را انتخاب می‌کنید و به چه دلیل؟ و کدام یک از مدل‌های فرایند نرم افزاری را انتخاب می‌کنید و به چه دلیل؟

۱۰-۳ باتوجه به شکل ۱-۳ که براساس نتایج بدست آمده از مطالعات علمی می‌باشد، به نظر می‌رسد که اسناد سودمندتر از اندازه‌ها و مقادیر می‌باشند. به نظر شما چرا چنین مسأله‌ای پدید آمده است و برای انتقال نقطه داده‌های اسنادی به دارای خط پسرقت (رگرسیون) در درون این نمودار چه کاری می‌توان انجام داد؟ یعنی برای ایجاد پیشرفت و اصلاح مقدار تعیین شده اسناد چه کاری می‌توان انجام داد؟

۱۱-۳ از شما خواسته شده است که یک عملیات (کار) کوچک که هر یک از دوره‌های پیشنهاد شده از سوی دانشگاه را تجزیه و تحلیل می‌کند، گسترش دهید و میزان متوسط کسب شده در هر رتبه را گزارش کنید. صورت وضعیتی از دامنه که این مشکل را محدود می‌کند، یادداشت نمایید.

۱۲-۳ سطح اولیه تجزیه و تحلیل عملی عملکرد صفحه‌آرایی را که به طور خلاصه در قسمت ۲-۳-۲

بیان شد را انجام دهید.

فهرست منابع و مراجع

- [AIR99] Airlie Council, "Performance Based Management: The Program Manager's Guide Based on the 16-point Plan and Related Metrics," Draft Report, March 8, 1999.
- [BAK72] Baker, F. T., "Chief Programmer Team Management of Production Programming," *IBM Systems journal*, vol. II, no. 1, 1972, pp. 56-73.
- [BOE96] Boehm, B., "Anchoring the Software Process," *IEEE Software*, vol. 13, no. 4, July 1996, pp. 73-82.
- [CON93] Constantine, L., "Work Organization: Paradigms for Project Management and Organization, *CACM*, vol. 36, no. 10, October 1993, pp. 34-43.
- [COU80] Cougar, J. and R. Zawacki, *Managing and Motivating Computer Personnel*, Wiley, 1980.
- [CUR88] Curtis, B. et al., "A Field Study of the Software Design Process for Large Systems," *IEEE Trans. Software Engineering*, vol. SE-31, no. II, November 1988, pp. 1268-1287.
- [CUR94] Curtis, B., et al., *People Management Capability Maturity Model*, Software Engineering Institute, 1994.
- [DEM98] DeMarco, T. and T. Lister, *Peopleware*, 2nd ed., Dorset House, 1998.
- [EDG95] Edgemon, J., "Right Stuff: How to Recognize It When Selecting a Project Manager," *Application Development Trends*, vol. 2, no. 5, May 1995, pp. 37-42.
- [FER98] Ferdinandi, P.L., "Facilitating Communication," *IEEE Software*, September 1998, pp. 92-96.
- [JAC98] Jackman, M., "Homeopathic Remedies for Team Toxicity," *IEEE Software*, July 1998, pp. 43-45.
- [KRA95] Kraul, R. and L. Streeter, "Coordination in Software Development," *CACM*, vol. 38, no. 3, March 1995, pp. 69-81.
- [MAN81] Mantei, M., "The Effect of Programming Team Structures on Programming Tasks," *CACM*, vol. 24, no. 3, March 1981, pp. 106-113.
- [PAG85] Page-Jones, M., *Practical Project Management*, Dorset House, 1985, p. vii.
- [REE99] Reel, J.S., "Critical Success Factors in Software Projects," *IEEE Software*, May, 1999, pp. 18-23.
- [WE186] Weinberg, G., *On Becoming a Technical Leader*, Dorset House, 1986.
- [WIT94] Whitaker, K., *Managing Software Maniacs*, Wiley, 1994.
- [ZAH94] Zahniser, R., "Timeboxing for Top Team Performance," *Software Development*, March 1994, pp. 35-38.

خواندنیهای دیگر و منابع اطلاعاتی

An excellent four volume series written by Weinberg (*Quality Software Management*, Dorset House, 1992, 1993, 1994, 1996) introduces basic systems thinking and management concepts, explains how to use measurements effectively, and addresses "congruent action," the ability to establish "fit" between the manager's needs, the needs of technical staff, and the needs of the business. It will provide both new and experienced managers with useful information. Brooks (*The Mythical Man-Month*, Anniversary Edition, Addison-Wesley, 1995) has updated his classic book to provide

new insight into software project and management issues. Purba and Shah (*How to Manage a Successful Software Project*, Wiley, 1995) present a number of case studies that indicate why some projects succeed and others fail. Bennatan (*Software Project Management in a Client/Server Environment*, Wiley, 1995) discusses special management issues associated with the development of client/server systems.

It can be argued that the most important aspect of software project management is people management. The definitive book on this subject has been written by DeMarco and Lister [DEM98], but the following books on this subject have been published in recent years and are worth examining:

- Beaudouin-Lafon, M., *Computer Supported cooperative Work*, Wiley-Liss, 1999.
 Carmel, E., *Global Software Teams: Collaborating Across Borders and Time Zones*, Prentice Hall, 1999.
 Humphrey, W.S., *Managing Technical People: Innovation, Teamwork, and the Software Process*, Addison-Wesley, 1997.
 Humphrey, W.S., *Introduction to the Team Software Process*, Addison-wesley, 1999.
 Jones, P.H., *Handbook of Team Design: A Practitioner's Guide to Team systems Development*, McGraw-Hill, 1997.
 Karolak, O.S., *Global Software Development: Managing Virtual Teams and Environments*, IEEE Computer Society, 1998.
 Mayer, M., *The Virtual Edge: Embracing Technology for Distributed Project Team Success*, Project Management Institute Publications, 1999.

Another excellent book by Weinberg [WEI86] is must reading for every project manager and every team leader. It will give you insight and guidance in ways to do your job more effectively. House (*The Human Side of Project Management*, Addison-Wesley, 1988) and Crosby (*Running Things: The Art of Making Things Happen*, McGraw-Hill, 1989) provide practical advice for managers who must deal with human as well as technical problems.

Even though they do not relate specifically to the software world and sometimes suffer from over-simplification and broad generalization, best-selling "management" books by Drucker (*Management Challenges for the 21st Century*, Harper Business, 1999), Buckingham and Coffman (*First, Break All the Rules: What the World's Greatest Managers Do Differently*, Simon and Schuster, 1999) and Christensen (*The Innovator's Dilemma*, Harvard Business School Press, 1997) emphasize "new rules" defined by a rapidly changing economy. Older titles such as *The One-Minute Manager* and *In search of Excellence* continue to provide valuable insights that can help you to manage people issues more effectively.

A wide variety of information sources on software project issues are available on the Internet. An up-to-date list of World Wide Web references that are relevant to the software projects can be found at the SEPA Web site:

[http://www.mhhe.com/engcs/compsci/pressman/resources/
project-mgmt.mhtml](http://www.mhhe.com/engcs/compsci/pressman/resources/project-mgmt.mhtml)

این کتاب تنها به خاطر حل مشکل دانشجویان پیام نور تبدیل به پی دی اف شد. همین جا از ناشر و نویسنده و تمام کسانی که با افزایش قیمت کتاب ما را مجبور به این کار کردند و یا متحمل ضرر شدند عذرخواهی می‌کنم.
گروهی از دانشجویان مهندسی کامپیوتر مرکز تهران

