بسم الله الرحمن الرحیم

فصل بیست و چهارم :

مفاهیم مدیریت پروژه

استاد مربوطه :

دکتر سید علی ابراهیمی رضوی

گردآورنده :

فاطمه شکاری بادی

شماره دانشجویی:

۹۸۰۱۶۴۵۶۹

مدیریت پروژه های نرم افزاری

در این بخش از کتاب،فنون مدیریتی موزد نیاز برای برنامه ریزی،پایش و کنترل پروژه های نرم افزاری را فراخواهید گرفت.در فصل هایی که به دنبال می آید،به این پرسش ها خواهیم پرداخت:

- افراد،فرایند و مساله را چگونه باید طی یک پروژه ی نرم افزاری مدیریت کرد؟
- معیار های نرم افزاری را چگونه میتوان در مدیریت یک پروژه ی نرم افزاری و غارینده نرم افزاری به کار گرفت؟
- تیم نرم افزاری چگونه برآوردای قابل اطمینانی از تلاش،هزینه و طول مدت پروژه به عمل می آورد؟
- برای ارزیابی ریسک هایی که میتوانند بر موقعیت پروژه تاثیر بگذارند، از چه فنونی میتوان استفاده کرد؟
- مدیر پروژه نرم افزاری چگونه یک مجموعه وظایف کاری برای مهندسی نف انتخاب میکند؟
- زمان بندی پروژه چگونه ایجاد میشود؟
- چرا نگهداری و مهندسی مجدد،هم برای مدیران و هم دست اندرکاران منف،این همه اهمیت دارد؟

هنگامی که به این پرسش ها پاسخ گفته شد،برای مدیریت پروژه های نرم افزاری،آماده تر خواهید بود که محصولی با کیفیت بالاتر را در زمان مقرر تحویل دهید.

به این پرسش ها پاسخ داده شد،مباحثی را خواهید آموخت که ممکن است در سال های آینده تاثیری عمیق بر مهندسی نرم افزار داشته باشد

مفاهیم مدیریت پروژه

نگاهی گذرا

مدیریت پروژه نرم افزاری چیست؟گرچه بسیاری از ما(در لحظات سیاه تر زندگی خود)به مدیریت به همان دیده ای نگریسته ایم که دیلبرت می نگرد،هنگام ساخت سیستم های کامپیوتری،مدیریت همچنان فعالیست بسیار ضروری باقی می ماند.مدیرت پروژه شامل برنامه ریزی،پایش و کنترل افراد،فرایند و رویداد هایی میشود که به موازات تکامل نرم افزار از یک مفهوم مقدماتی به استقرار عملیاتی کامل رخ می دهند.

چه کسی آن را انجام میدهد؟هرکسی تا اندازه ای«مدیریت میکند»ولی حوزه ی فعالیت های مدیریتی در میان افراد درگیر در یک پروژه ی نرم افزاری متفاوت است.مهندس نرمافزار،فعالیت های روزانه،برنامه ریزی،پایش وکنترل میکنند.مدیران ارشد،ارتباط میان مدیران نرم افزار و مدیران نرم افزار و مدیران تجاری را هماهنگ میکنند.

چرا اهمیت دارد؟ساخت نرم افزار کامپیوتری کاری پیچیده است،به ویژه اگر شامل تعداد بسیاری از افراد شود که در مدتی نسبتا طولانی مشغول به کار باشند.از همین رو است که پروژه های نیاز به مدیرت دارند.

مراحل کار کدام است؟درک اهمیت این چهارعامل:افراد،محصول،فرایند و پروژه.افراد را باید به گونه ای سازماندهی کرد که کار نرم افزاری را بطور اثربخش انجام دهند.برقراری ارتباط با مشتری و سایر طرف های قرارداد ذی نفع باید به گونه ای رخ دهد که حوزه و نیازمندی های سیستم درک شوند.باید فرایندی انتخاب شود که افراد و محصول،مناسب باشد.پروژه باید با برآورد نیروی کار و زمان تقویمی لازم جهت انجام وظایف کاری،برنامه ریزی شود:تعریف محصولات کاری،با نهادن نقاطی برای چک کردن کیفیت(نقاط بازرسی)و شناسایی سازوکارهایی برایاش و کنترل کار تعریف شده در برنامه ریزی از مراحل کارهستند

محصول کاری چیست؟باشروع فعالیت های مدیریتی،یک برنامه ریزی پروژه ایجاد میشود.دراین طرح،فرایند و وظایفی که قرار است اجرا شود،افرادی که کار را انجام میدهند و سازوکارهایی برای ارزیابی ریسک ها،کنترل تغییرات و ارزیابی کیفیت دراین طرح تعریف میشوند.

چگونه مطمئن شوم که درست از عهده ی انجام کار برآمده ام؟هرگز به طور کامل نمیتوان اطمینان یافت که برنامه ریزی پروژه درست است تااین که یک محصول باکیفیت بالا و سروقت درچارچوب بودجه ی تعیین شده تحویل دهید.به هرحال،مدیر پروژه هنگامی کار خود را درست انجام داده است که زیردستان خودرا به کار گروهی اثربخش تشویق کند،به طوری که توجه خود رابه نیازهای مشتری و کیفیت محصول معطوف سازد.

۱-۲۴طیف مدیریتی

مدیریت پروژه ی نرم افزاری برچهار مورد تکیه دارد:افراد،محصول،فرایند و پروژه.ترتیب این موارد اختیاری نیست.مدیری که فراموش کند کار مهندسی نرم افزار،یک تلاش کاملا انسانی است،هرگز درمدیریت پروژه موفق نخواهد شد. مدیری که درهمان مراحل آغازین تکامل محصول ازبرقرای ارتباط فراگیر با طرف های ذی نفع غافل بماند،این ریسک را به جان میخرد که برای مشکلی غیر از آن چه مورد نظر آن هاست،راهکار پیدا کند.مدیری که به فرایند توجه چندانی نمیکند،این ریسک را ایجاد میکند که روش ها و ابزارهای فنی رقیب در کنار هم قرار میگیرند.مدیری که بدون یک،برنامه ریزی مستحکم شروع به کار کند،موفقیت محصول را به خطر می اندازد.

۱-۱-۲٤ افراد

پرورش آدم های نرم افزاری پرانگیزه و با مهارت بالا از دهه ی ۱۹۶۰ مورد بحث بوده است.در واقع،«عامل انسانی»چنان مهم است که موسسه ی مهندسی نرم افزاری یک مدل بلوغ قابلیت های انسانی تدارک دیده است که پاسخی است به این واقعیت که«هرسازمانی باید پیوسته توانایی خود را در جذب،توسعه،انگیزش،سازمان دهی و حذف نیروی کارمورد نیاز برای پیشبرد اهداف تجاری راهبردی خود بهبود بخشد.»

مدل بلوغ قابیلت های انسانی،چند زمینه ی عملیاتی کلیدی برای آدم های نرم افزار تعریف میکند.که عبارتند از:تعیین پرسنل،برقراری ارتباط و هماهنگی،محیط کار،مدیریت کارایی،آموزش،جبران،تحلیل و توسعه ی رقابت،توسعه ی کاری،توسعه کارگروهی،توسعه تیمیوغیره.درسازمان هایی که به سطوح بالایی از مدل بلوغ قابلیت های انسانیدست پیدا میکنند،احتمال پیاده سازی اثربخش کارهای مدیریت پروژه ی نرم افزاری بسیار بیشتراست.

مدل بلوغ قابلیت های انسانی در کنار مدل انسجام بلوغ قابلیت های نرم افزاری(فصل ۳۰)سازمان ها را در ایجاد یک فرایند نرم افزاری بالغ یاری میدهد.به مسائل مرتبط با مدیریت افراد و ساختار برای پروژه های نرم افزاری بعداً در همین فصل خواهیم پرداخت.

۲-۱-۲٤ محصول

پیش از آنکه بتوان برای پروژه برنامه ریزی کرد،اهداف و حوزه ی محصول باید تعیین شود،راهکارهای متفاوت باید مدنظر قرار گیرد و قید و بند های مدیریتی و فنی باید شناسایی شود.بدون این اطلاعات،تعریغ منطقی(وصحیح)برآورد هزینه،ارزیابی اثربخش ریسک ها،تقسیم واقع بینانه ی وظایف یا زمان بندی قابل مدیریت پروژه که شاخصی با معنی از پیشرفت کاربدهد،امکان پذیر نخواهد بود.

شما،به عنوان سازنده ی نرم افزار و سایر طرف های ذی نفع باید با یکدیگرملاقات کنید تا اهداف و حوزه ی محصول راتعیین کنید.دربسیاری موارد،این فعالیت به عنوان بخشی از مهندسی سیستم و مهندسی خواسته های سیستم آغار میشود وبه عنوان نخستین مرحله در مهندسی خواسته های سیستم(فصل۵)ادامه می یابد.اهداف کلی محصول(ازدیدگاه ذی نفغ ها)بدون درنظر گرفتن چگونگی دستیابی به آن ها مشخص می شوند.در داخل حوزه،داده های اولیه،قابلیت های عملیاتی و رفتارهایی که محصول رامشخص می کنند و مهمتر از آن،تلاش های به عمل آمده برای محصورکردن این خصوصیات به شیوه های کمی،شناسایی می شود.

هنگامی که اهداف و حوزه ی محصول شناسایی شد،راهکارهای متفاوت،مدنظر قرار می گیرد.گرچه درباره ی جزئیات بسیار اندکی بحث میشود،این راهکارهای متفاوت به مدیران و دست اندرکاران این امکان را میدهد که با عنایت به قید و بندهای ناشی از مهلت تحویل،محدودیت های بودجه ای،قابلیت دسترسی به پرسنل،رابط های فنی و بسیاری از عوامل دیگر،«بهترین»رویکرد را انتخاب کنند.

۳-۱-۲٤ فرایند

فرایند نرم افزاری(فصل های۲و۳)چارچوبی فراهم می سازد که براساس آن میتوان یک طرح جامع برای توسعه ی نرم افزار ارائه داد.تعداد اندکی از فعالیت ها چارچوبی وجود دارند که فارغ از اندازه یا پیچیدگی شان برای تمامی پروژه هاینرم افزاری به کار میروند.چند مجموعه وظیفه ی متفاوت(وظایف،نقاط عطف،محصولات پروژه ی نرم افزاری و خواسته های تیم پروژه تطابق یابند.سرانجام،فعالیت های چتری(نظیم تضمین کیفیت نرم افزار،مدیریت پیکربندی نرم افزار و اندازه گیری)برمدل فرایندی سوار میشوند.فعالیت های چتری از هرگونه فعالیت چارچوبی مستقل هستند و در سرتاسر فرایند روی میدهند.

٤-۱-۲٤ پروژه

ما پروژه های نرم افزاری برنامه ریزی شده و کنترل شده را به یک دلیل اصلی اجرا می کنیم(چون این تنها راه مدیریت پیچیدگی هاست)و درعین حال،تیم های نرم افزاری هنوز درحال تقلا هستند.طی مطالعه ای که بین سال های ۱۹۹۸تا۲۰۰۴روی ۲۵۰پروژه ی نرم افزاری بزرگ انجام شده است،کاپرزجونز[jon04]دریافته است که«نزدیک به ۲۵ شرکت از لحاظ دستیابی به اهداف کیفیتی در زمان و هزینه تامین شده موفق بوده اند.حدود ۵۰ پروژه تاخیر یا کسر بودجه ی زیر۳۵٪داشته اند در حالی که ۱۷۵پروژه شاهد تاخیر های جدی و کسر بودجه های چشمگیر بوده اند یا حتی نتوانسته اند محصول را کامل کنند .»گرچه آهنگ موفقیت برای پروژه های نرم افزاری کنونی قدری بهبود یافته اند هنوز از آنچه که باید باشد،بسیار بیشتر است.

برای پرهیز از شکست پروژه یک مدیر پروژه ی نرم افزاری و مهندسان نرم افزی که محصول را میسازند باید از یک سری علائم هشدار دهنده رایج پرهیز کنند ،عوامل مهم در موفقیت را که به مدیریت خود پروژه منجر میشود، بشناسند و یک رویکرد مبتنی بر عقل سلیم برای برنامه ریزی،پایش و کنترل پروژه داشته باشند. هرکدام از این مسائل دربخش ۵-۲۴و فصل های آتی مورد بحث قرار خواهیم داد.

۲-۲۴ افراد

طی مطالعه ای که توسطIEEEمنتشر شده است(Cur88)،ازمعاونان مهندسی سه شرکت فن آوری بزرگ پرسیده شد که مهمترین عامل در موفقیت یک پروژه نرم افزاری چیست.پاسخ آن ها به شرح زیر بود:

معاون۱:به گمان من اگر قرار باشد یک چیز را به عنوان مهمترین عامل در محیط خودمان انتخاب کنیم،میگوییم آن یک چیز ابزارها نیست بلکه افراد است.

معاون۲:مهمترین عامل موفقیت ما در این پروژه،داشتن آدم های زرنگ بود...به نظر من موارد دیگر خیلی اهمیت ندارند...مهمترین کاری که برای پروژه باید انجام بدهید،گزینش پرسنل است...موفقیت سازمان توسعه نرم افزار،خیلی،خیلی به توانایی جمع کردن ادم های درست بستگی دارد.

معاون۳:تنهای قانونی که من دارم،حصول اطمینان از داشتن آدم های درست است(آدم های های واقعا درست)و این که آدم های درست تربیت کنم(و این که محیطی فراهم کنم که آدم های درست در آن بتوانند به کار و تولید مشغول شوند).

درواقع،این یک بیانیه ی قانع کننده درباره ی اهمیت افراد در فرایند مهندسی نرم افزار است.بااین حال،همه ی ما از معاون شرکت، که مهندس ارشد است،تا دون پایه ترین کارمندان،قالبا آن گونه که باید،قدردان نیروی انسانی نیستند.مدیران چنین استدلال میکنند که پرسنل در درجه اول اهمیت قرار دارند ولی آنچه در عمل انجام می دهند،خلاف گفته هایشان است.دراین بخش به بررسی ذی نفع هایی خواهیم پرداخت که در فرایند نرم افزار شرکت دارند و شیوه ی سازماندهی آنها برای اجرای مهندسی نرم افزار اثر بخش را مورد بحث قرار خواهیم داد.

۱-۲-۲٤ طرف های دی نفع

فرایند نرم افزار(و هر پروژه ی نرم افزاری)شلمل مجموعه ای از طرف های ذی نفع میشود که می توان آن ها را به پنج گروه تقسیم کرد:

۱.مدیران ارشد که به تعریف مسائل تجاری ای می پردازند که غالباً تاثیر چشمگیر بر پروژه دارند.

۲.مدیران پروژه(فنی)که باید برای دست اندرکاران نرم افزار برنامه ریزی،ایجاد انگیزه،سازمان دهی و آن ها را کنترل کنند.

۳.دست اندرکاران که واجد مهارت های فنی لازم برای مهندسی یک محصول یا برنامه ی کاربردی هستند.

۴.مشتریان که خواسته های محصول را مشخص می کنند و سایر ذی نفع هایی که به نتیجه کار توجه نشان می دهند.

۵.کاربران نهایی که پس از ارائه محصول با آن تعامل دارند.

هر پروژه ی نرم افزاری شامل یک سریع افراد میشود که در این طبقه بندی می گنجد.تیم پروژه برای این که موثر واقع شود،باید به گونه ایسازمان دهی شود که مهارت ها و توانایی های تک تک افراد را به حداکثر برساند.و این وظیفه رهبر تیم است.

۲-۲-۲٤ رهبر تیم

مدیریت پروژه یک فعالیت گروهی است و از این رو آنهایی که روحیه ی رقابتی دارند.رهبر خوبی برای تیم پروژه نمیشوند.به زبان ساده،فاقد آمیزه ی درستی از مهارت های انسانی هستند.بااین وجود،اجمون میگوید:«متاسفانه،وبه وفور،به نظر می رسد آدم ها صرفاً بر حسب تصادف مدیر پروژه میشوند.[Edg95]

جری واینبرگ در یک کتاب عالی درباب رهبری فنی[Wei86]یک مدل MOIپیشنهاد میکند:

انگیزش(Motivation).توانایی ترغیب آدم های فنی به بروز بهترین توانایی های آنها.

سازمان دهی(Organization).توانایی قالب دادن به فرایند های موجود(یا ابداغ فرایند های جدید)که تبدیل مفهوم اولیه به محصول نهایی را میسر کنند.

ایده ها یا نوآوری(Ideas or Innovation) توانایی تشویق افراد به خلاقیت و ایجاد حس خلاقیت،حتی هنگامیکه باید در محدوده قیدوبندهای تعیین شده برای یک محصول نرم افزاری خاص کار کنند.

وینبرگ معتقد است که رهبران پروژه های موفق از شیوه های حل مساله برای مدیریت استفاده می کنند.یعنی،مدیر پروژه نرم افزاری باید توجه خود را به درک مساله ای که قراراست حل شود،معطوف کند.جریان ایده هارا مدیریت کند و در عین حال همه‌ی افراد تیم را آگاه کند(در گفتار و بسیار مهم تراز آن،در عمل)که کیفیت حرف اول را میزند وجای مسnاحمه ندارد.

در یک دیدگاه دیگر [edg95]در مورد خصوصیات مدیریت اثر بخش پروژه ،بر چهار ویژگی کلیدی تاکید می کند:

حل مسئله (problem solving): مدیر کار امد پروژه نرم افزاری می تواند مشکلات سازمانی و فنی مرتبط با پروژه را تایین می کند. به طور سیستماتیک یک راهکار ارائه کند یا بطور مناسب سایر دست اندرکاران را برای ارائه راهکار انگیزش کند. درس های اموخته از پروژه های پیشین را برای شرایط جدید به کار گیرد و انعطاف پذیری کافی برای تغییر جهت گیری در صورت بی ثمر بودن راهکار را داشته باشد .

هویت مدیریتی(managerial identity) :مدیر خوب باید ابتکار عمل پروژه را در دست داشته باشد. او باید از اعتماد به نفس لازم برای کنترل اوضاع برخوردار باشد و ترتیبی اتخاذ کند که افراد خوب تیم بتوانند از غریزه ی خود پیروی کند.

موفقیت(achievement): مدیر خوب باید به ابتکارها و نوآوری های پاداش دهد تا بهره وری تیم پروژه بالا رود . او باید از طریق کنش های خود نشان دهد که خطرپذیری کنترل شده به تنبیه منجر نخواهد شد.

تاثیر گذاری و تشکیل تیم(influence and team building): یک مدیر پروژه کارامد باید بتواند ذهن افراد را بخواند : او باید قادر به درک علامت های لفظی یا غیر لفظی باشد و به نیازهای افرادی که این علامت ها را ارسال می کنند واکنش نشان دهد.

۲۴-۲-۳ تیم نرم افزاری

به تعداد سازمان هایی که نرم افزار می سازند، ساختارهای سازمانی برای نیروی انسانی وجود دارد ، خوب یا بد ، ساختار سازمانی به راحتی قابل اصلاح نیست دغدغه های مرتبط با پیامدهای عملی و سیاسی تغییرات سازمانی ، در حوزه مسئولیت های مدیر پروژه های نرم افزاری نیستند. به هرحال سازماندهی افرادی که به طور مستقیم در یک پروژه ی نرم افزاری جدید شرکت دارند ، در قلمرو مدیر پروژه است .

«بهترین» ساختار تیمی ، به سبک مدیریتی زمان شما ، تعداد ادمهایی که تیم را تشکیل می دهند و سطوح مهارتی انها و البته به سطح مشکل بودن مسئله بستگی دارد. مالتی(man81) هفت عامل پروژه ایی را توصیف می کند که هنگام برنامه ریزی برای ساختار تیم های مهندسی نرم افزار باید مدنظر داشت:

- مشکل بودن مساله ایی که قرار است حل شود.
- «اندازه» برنامه های حاصل بر حسب تعداد خطوط کد نوشته شده یا نقاط عملکرد
- مدت زمانی که اعضای تیم کنار یک دیگر می مانند(طول عمر تیم)
- میزان قابلیت مساله برای پیمانه بندی
- کیفیت و قابلیت اطمینان لازم برای سیستمی که قرار است ساخته شود

- میزان قطعی بودن تاریخ تحویل
- درجه ی قابلیت برقراری ارتباط در پروژه

کنسانتین (con93) چهار <<الگوی سازمانی>> برای تیم های مهندسی نرم افزار پیشنهاد می کند:

1. الگوی بسته،تیم را بر اساس یک سلسله مراتب سنتی ساختاردهی میکند.اینگونه تیم ها هنگامی میتوانند خوب کار کنند که پروژه کاری آن ها کاملا مشابه باتلاش های گذشته آن ها باشد،ولی درصورت کار در یک الگوی بسته،احتمالا نوآوری کمتر خواهد بود.

2. الگوی تصادفی،تیم را آزادانه ساختار دهی میکندو به ابتکار تک تک اعضای تیم وابسته است.در صورت نیاز به ابتکار و نوآوری در امور فنی،تیم هایی که از الگوی تصادفی پیروی میکنندموفق خواهند بود،ولی اینگونه تیم ها درصورت نیاز به (عملکرد منظم)ممکن است به تقلا بیوفتند.

3. در الگوی باز تلاش میشود تا تیم به گونه ای ساختاردهی شود که به برخی کنترل های مرتبط باالگوی بسته دستیابی داشته باشد و درعین حال بسیاری از نوآوری هایی که در الگوی تصادفی رخ میدهدنیز امکان پذیرباشد.انجام کار بصورت گروهی،ارتباطات در سطحی گسترده و تصمیم گیری براساس اجماع و اتفاق نظر،از شاخصه های بارز الگوی باز به شمار میرود.ساختارهای تیمی مبتنی بر الگوی باز برای حل مسائل پیچیده بسیار مناسب هستند،ولی ممکن است به اندازه سایر تیم ها اثربخش نباشند.

4. الگوی همگام بر قطعه قطعه کردن یک مساله به شیوه ای طبیعی و سازماندهی اعضای تیم برای کار روی هریک ازاین قطعات تاکید دارد،به گونه ای که میان اعضای تیم ارتباط چندانی برقرار نیست.

به عنوان یک حاشیه تاریخی،خوب است اشاره شود که یکی از قدیمی ترین سازمان هابرای تیم های نرم افزاری،ساختار الگوی بسته ای بود که ابتدا تیم برنامه نویس ارشد نامیده میشد،این ساختار را نخست هارلان میلز پیشنهاد کرد و بیکر[Bak72]آن را توصیف کرد:

هسته تیم شامل یک مهندس ارشد(برنامه نویس ارشد)که برای برنامه ریزی،هماهنگی و مرور کلیه فعالیت های تیم،کارمندان فنی(معمولا دوتا پنج نفرند)بدای تحلیل و توسعه فعالیت ها و مهندس پشتیبان برای یاری رساندن به مهندس ارشد در انجام فعالیت هایش میشود که میتواند در تداوم پروژه باحداقل زبان جایگزین مهندس ارشد شود.ممکن است یک یاچند متخصص(مثلا کارشناس مخابرات یا طراح)و چند کارمند پشتیبان(مانند نویسندگان فنی و پرسنل اداری)در خدمت مهندس ارشد باشند.

به عنوان نقطه مقابل ساختار تیمی برنامه نویس ارشد،در الگوی تصادفی کنستانتین[Con93]یک تیم نرم افزاری با استقلال در نوآوری پیشنهاد میشودکه رویکرد آن هارا در قبال کار،شاید بتوان به بهترین نحوبا عبارت هرج و مرج نوآورانه توصیف کرد.گرچه این رویکرد آزادپروری،جاذبه هایی دارد،هدایت انرژی نوآوری به درون تیمی باکارایی بالا باید هدف اصلی سازمان مهندسی نرم افزارباشد.برای دستیابی به تیمی باکارایی بالا:

- اعضای تیم باید به هم اعتماد کنند.
- توزیع مهارت ها باید با مساله متناسب باشد.
- اگر قراربباشدیکپارچگی تیم حفظ شود،شاید لازم باشد آدم های تک‌رو از تیم طرد شوند

سازماندهی تیم هرچه که باشد،هدف هرمدیر پروژه کمک به ایجاد تیمی است که از خود یکپارچگی نشان دهد.دومارکو و لیستردر کتاب خود[Dem98]مساله رااینگونه مورد بحث قرار میدهند:

ماذاتا از واژه ی تیم دردنیای تجارت،نسبتا آزادانه استفاده میکنیم و به هر گروهی که به آن ها کاری محول شود،واژه(تیم)را اطلاق میکنیم.ولی بسیاری ازاین گروه ها،تیم حساب نمیشوند.آن ها تعریف مشترکی از موفقیت ندارند.و یک روح تیمی در آن ها دیده نمیشود،چیزی که جای آن خالی است،پدیده ای است که آن را ژله میخوانیم.

تیم ژله ای به گروهی از افراد گفته میشودکه طوری باهم پیوند یافته اند که کل آن ها از مجموع اعضابزرگ تر است...

هنگامیکه تیمی شروع به ژله بندی میکند،احتمال موفقیت به مراتب بیشتر میشود،این تیم دیگر توقف ناپذیر است و مثل کماندوها برای موفقیت به پیش میتازد...نیازی به مدیریت آن ها به شیوه های سنتی نیست و قطعا نیازی به انگیزش ندارند.آن ها خودشان از انگیزه برخوردارند.دومارکو و لیستر معتقدند که تیم های ژله ای بسیار بیش ازمیزان میانگین، بهره وری و انگیزه دارند.

هدف آن ها مشترک است،فرهنگ مشترکی دارند و در بسیاری موارد،یکجور نخبگی دارند که آن هارا از دیگران متمایز میکند.به هرحال،همه تیم ها ژله بندی نمی کنند.درواقع تیم ها به مشکلاتی مبتلا هستندکه جکمن آن هارا(سموم تیم)مینامد.او پنج عامل را برمیشمرد که به سمی تر شدن تیم کمک میکنند:(۱)فضای کاری دیوانه وار،(۲)فضای ملتهبی که باعث ایجاد اصطحکاک میان اعضای تیم میشود،(۳)فرایند نرم افزاری که از هماهنگی خوبی برخوردار نیست یا پاره پاره است،(۴)تعریف مبهمی از نقش ها درتیم نرم افزاری و(۵)تداوم و تکرار شکست.

برای پرهیز از یک محیط کاری (دیوانه وار) مدیر پروژه باید یقین حاصل کند که تیم او به تمام اطلاعات لازم برای انجام وظایف خود دستیابی داردو هنگامیکه اهداف اصلی تعیین شدند ،دیگر نباید تغییر داده شوند،مگر اینکه تغییر مطلقا ضروری باشد. تیم نرم افزاری میتواند از التهاب دوری کند مشروط برآن‌که حداکثر مسولیت ممکن برای تصمیم گیری ها به آن محول شود.یک فرایند نامناسب(مثلا وظایف کاری بیهوده یاسنگین)باشناخت درست از محصولی که قراراست ساخته شود ،افرادی که کاررا انجام میدهند و آزادگذاشتن تیم در گزینش مدل فرایندی،قابل پرهیز است. برای دستیابی به این منظور هستند)ارائه دهد و یک سری اقدام های تصحیحی برای عملکرد نادرست اعضای تیم تعیین کند و سرانجام اینکه،کلید پرهیز از یکجور شکست و ناکامی،بنا نهادن تکنیک های تیم محور برای دریافت بازخورد و حل مساله است.

علاوه برا پنج سمی که جکمن شرح میدهد، تیم نرم افزاری غالبا باعادت های انسانی متفاوت اعضای تیم هم دست به گریبان است.بعضی ها درونگرا هستند،درحالی که عده ای دیگر برونگراترند.عده ای هستند که اطلاعات را با هوشمندی جمع آوری می کندد و حقایق نامتجانس،مفاهیم گسترده تری استخراج می کنند.عده ای،اطلاعات را به طور خطی پردازش می کنند و از داده های ارائه شده جزئیات دقیقی را جمع آوری و سازماندهی می کنند.برخی از اعضای تیم تنها درصورتی میتوانند تصمیم گیری کنند که بایک استدلال منطقی و منظم مواجه باشند.عده ای دیگر براساس احساسات خویش تصمیم گیری می کنند.

برخی دوست دارند یک برنامه ی زمان بندی شده آکنده از وظایف سازمان دهی شده داشته باشند که آنهارا به دستیابی به بخشی از اهداف پروژه قادر سازد.سایرین محیطی با خودمختاری بیشتر را ترجیح میدهند در آن قدری آزاد عمل داشته باشند.برخی به سختی کار میکنند تا کارهارا مدت ها پیش از رسیدن به تاریخ تایین شده انجام دهند و به ترتیب از فشار های ناشی از فرا رسیدن تاریخ تحویل بکاهند،درحالی که عده ای دیگر،همه چیز را به دقیقه ی نود موکول میکنند.بحث مفصلی در خصوص این صفات شخصی و شیوه های همساز کردن افرادی با صفحات متضاد در یک تیم نرم افزاری از حوصله ی این کتاب خارج است.به هرحال،شایان ذکر است که اشراف به یک تفاوت های انسانی،گام نخست در ایجاد تیمی است که توان ژله بندی داشته باشد.

۲۴-۲-٤ تیم های چابک

طی دهه ی اخیر،توسعه چابک نرم افزاری(فصل۳)به عنوان راهکار برای بسیاری از مشکلات پروژه های نرم افزاری پیشنهاد شده است.یاد اوری میکنیم که فلسفه ی چابکی،مشوق مشتری مداری و تحویل زودهنگام اولین نسخه های نرم افزار،تیم های کوچک با انگیزه ی بالا،روش های غیر رسمی،محصولات کاری کمینه و سادگی در توسعه است.

تیم های کوچک با انگیزه ی بالا حالا که تیم چابک نامیده میشوند،خصوصیات تیم های نرم افزاری بحث شده در بخش قبل را دارند و از سموم بسیاری که ایجاد مشکل میکندد،پرهیز دارند.به هرحال،در فلسفه ی چابکی،رقابت فردی(عضو تیم)در کنار همکاری به عنوان عوامل مهم موفقیت مورد تاکید قرار میگیرند.کاکبرن و های اسمیت[Coc0la]در این خصوص چنین مینویسند:

٤ تیم های چابک

طی دهه ی اخیر،توسعه چابک نرم افزاری(فصل۳)به عنوان راهکار برای بسیاری از مشکلات پروژه های نرم افزاری پیشنهاد شده است.یاد اوری میکنیم که فلسفه ی چابکی،مشوق مشتری مداری و تحویل زودهنگام اولین نسخه های نرم افزار،تیم های کوچک با انگیزه ی بالا،روش های غیر رسمی،محصولات کاری کمینه و سادگی در توسعه است.

تیم های کوچک با انگیزه ی بالا حالا که تیم چابک نامیده میشوند،خصوصیات تیم های نرم افزاری بحث شده در بخش قبل را دارند و از سموم بسیاری که ایجاد مشکل میکندد،پرهیز دارند.به هرحال،در فلسفه ی چابکی،رقابت فردی(عضو تیم)در کنار همکاری به عنوان عوامل مهم موفقیت مورد تاکید قرار میگیرند.کاکبرن و های اسمیت[Coc0la]در این خصوص چنین مینویسند:

برای استفاده ی اثر بخش از رقابت های موجود میان اعضای یک تیم و تقویت همکاری کارامد در یک پروژه نرم افزاری،تیم های چابک قادر به خود سازمان دهی هستند.تیم خود سازمان ده الزاماً به یک ساختار تیمی واحد بسنده نمیکند.

دربسیاری از مدل های فرایندی(مثلا اسکرام)به تیم چابک،خودمختاری چشمگیری در مدیریت پروژه و تصمیم گیری های فنی برای انجام کار داده میشود.برنامه ریزی در کمترین سطح ممکن انجام میشود و تیم مجاز است تا خودش رویکردی(مثلاً فرایند،روش ها،ابزارها)را برگزیند و تنها قید و بندهای موجود،خواسته های تجاری و استاندارد های سازمانی خواهند بود. باپیشرفت پروژه،تیم،خودش را به گونه ای سازماندهی میکند که در نقطه ی معینی از زمان،بیشترین بهره را به پروژه برساند.برای نیل به این مقصود،تیم چابک ممکن است جلسات روزانه برگزار کند تا برای کارهایی که قرار است در آن روز انجام شود،هماهنگی لازم به عمل آید.

تیم نرم افزاری براساس اطلاعات به دست آمده در این جلسات رویکرد خود را به گونه ای انتخاب میکند که بخشی از کار را پیش ببرد.با گذشت هرروزه،خودسازماندهی مداوم و همکاری،تیم را به سوی نسخه ی جدیدی از نرم افزار نزدیک میکند.

24-2-5 مسائل مربوط به هماهنگی و ارتباطات

دلایل فراوانی وجود دارد که پروژه نرم افزاری دچار مشکل شود .بسیاری از کارهای در مقیاس بزرگ انجام میشوند که این به پیچیدگی,سردرگمی واشکالات جدی در ایجاد هماهنگی میان اعضای تیم منجر میشود.عدم قطعیت ,اشکالی شایع است که نتیجه اش چریان مستمری از تغییرات است که حرکت پروژه را کند میکند . در بسیاری از سیستم ها قابلیت همکاری متقابل به یک خصوصیت کلیدی تبدیل شده است.نرم افزارهای جدید باید با نرم افزار های موجود ارتباط برقرار کنند و از قید و بندهای تحمیل شده از سوی سیستم یا محصول پیروی

کنند.این خصوصیات نرم افزار های مدرن(مقیاس,عدم قطعیت و قابلیت همکاری متقابل) حقایق غیر قابل انکارند.به منظور تقابل اثربخشی باآنها باید روش هایی اثربخش برای برقراری هماهنگی میان افراد گروه تدارک دید.برای این منظور ,سازوکارهایی جهت برقراری ارتباطات رسمی از طریق (مکاتبات,جلست رسمی و سایر کانال های ارتباطی نسبتا غیر تعاملی)برقرار میشود[kra95],ارتباطات غیر رسمی,شخص ترند,اعضای تیم نرم افزاری,ایده هارا به شیوه ای تک منظوره به اشتراک میگذارند.با بروز مشکل در خواست کمک میکنند و روزانه باهم درتعامل هستند.

۳-۲۴ محصول

مدیر پروژه ای نرم افزاری در همان ابتدای شروع پروژه بایک معضل بزرگ مواجه است.برآوردهای کمی و طرحی سازمان یافته مورد نیاز است ولی اطلاعات متقن در دسترس نیستند.تحلیل مشروح خواسته های نرم افزار اطلاعات لازم برای این براوردها را فراهم میسازد.ولی این تحلیل غالبا به چند هفته یا حتی چندماه زمان نیاز دارد.بدتر اینکه خواسته ها ممکن است سیال باشند و یا پیشرفت پروژه دائما تغییر کند.

خواه ناخواه باید مساله ای را که قرار است حل شود و نیز محصول را در همان اغاز پروژه بررسی کنید در یک سطح کمینه حوزه ی محصول باید معین و حد و مرز انها مشخص شوند.

۱-۳-۲٤حوزه ی نرم افزار

نخستین پروژه عملیاتی در یک پروژه نرم افزاری تعیین حوزه نرم افزار است.این جوزه با پاسخ دادن به پرسش های زیر قابل تعریف است.

حیطه:نرم افزاری که قرار است ساخته شود چگونه در یک سیستم بزرگتر محصول یاحیطه تجاری خواهد گنجید و در نتیجه این حیطه چه قید و بندهایی تحمیل میشوند؟

اهداف اطلاعاتی:اشیای داده ای که به عنوان خروجی نرم افزار تولید میشوند و برای مشتری قابل مشاهده هستند چیستند؟

عملکرد و کارایی:نرم افزار چه عملیاتی انجام میدهد تا داده های ورودی را به خروجی تبدیل کند؟ایا خصوصیات کارایی خاصی وجود دارد که باید به انها پرداخته شود؟

حوزه ی پروژه ی نرم افزاری باید عاری از هرگونه ابهام باشد و در سطوح مدیریتی و فنی بتوان آن را درک کرد.بیان حوزه ی نرم افزار باید مقید باشد.یعنی داده های کمی(مثلا تعداد کاربران همزمان،محیط هدف،حداکثر زمان پاسخ دهی مجاز)به صراحت بیان می شوند،قید و بندها و یا محدودیت ها(مثلا هزینه ی محصول،اندازه ی حافظه را محدود میکند)ذکر میشوند و عوامل تسکین دهنده(مثلا این که الگوریتم های مطلوب به خوبی در جاوا موجودند)توصیف میشوند.

۲-۳-۲٤ تجزیه ی مساله

نجزیه مساله،که گاهی آن را افرازبندی یا خردکردن نیز مینامند،فعالیتی است که هسته ی تحلیل خواسته های نرم افزار را تشکیل می دهد(فصل های۶و۷).طی فعالیت تعیین حوزه،هیچ تلاشی برای تجزیه ی کامل مساله به عمل نمی آید.درعوض،تجزیه در دو زمینه اصلی انجام میشود(۱)قابلیت عملیاتی و محتوا(اطلاعاتی)که باید تحویل شود(۲)فرایندی که برای تحویل آن بکار میرود.

انسان به طور ذاتی هنگام مواجهه با یک مشکل پیچیده به راهبرد تقسیم و غلبه روی می آورد.به زبان ساده،یک مساله ی پیچیده به مسائل ساده تری افراز میشود که بیشترقابل مدیریت هستند.این راهبردی است که با آغاز برنامه ریزی برای پروژه کاربرد دارد.عملکردهای نرم افزار،که در بیان حوزه توصیف می شوند،ارزیابی و پالایش میشوند.پیش از آغاز برآورد(فصل۲۶)جزئیات بیشتری فراهم شود. چون برآوردهای هم هزینه و هم زمانبندی،جهت گیری عملیتی دارند،غالباً میزانی از تجزیه مفید است.
بطور مشابه،اشیای داده ای یامحتوای اصلی،به اجزای سازنده شات تجزیه میشوند و درکی منطقی از اطلاعات تولید شده توسط نرم افزار فراهم مس آورند.

برای مثال،پروژه ای را در نظر بگیرید که محصول آن یک واژه پرداز جدید است.ازجمله ویژگی های منحصر به فرد این محصول،گفتار پیوسته،ورودی صفحه کلید مجازی از طریق صفحه لمسی،ویژگی های ویرایش-کپی خودکار کاملا پیچیده،قابلیت صفحه آرایی،استخراج خودکار نمایه و فهرست مندرجات،وغیره است.

مدیر پروژه نخست باید یک بیان حوزه تهیه کند که این ویژگی ها (و سایرویژگی های رایج نظیر ویرایش،مدیریت فایل،تولید سند)را مقید کند.برای مثال،آیا ورودی گفتاری به گونه ای است که کاربر باید محصول را برای آشنایی با صدای خودش «آموزش» دهد؟

بطور مشخص،ویژگی ویرایش-کپی چه قابلیت هایی را فراهم می آورد؟قابلیت صفحه آرایی تا چه حد پیچیده خواهد بود و آیا قابلیت های ناشی از صفحه لمسی را نیز در بر خواهد گرفت؟

با تکامل پیدا کردن بیان حوزه،سطح نخستی از افرازبندی طبیعتاًرخ میدهد.تیم پروژه در میابد که بخش بازاریابی با مشتریان بالقوه صحبت کرده است و فهمیده است که عملکرد های زیر باید بخشی از قابلیت ویرایش-کپی باشند:

(۱)اصلاح غلط های املایی،(۲)اصلاح غلط های دیتوری(۳)اصلاح ارجاعات برای مستندات بزرگ(مثلا این که آیا ارجاع به یک مدخل کتاب شناسی در فهرست منابع،درست هست یاخیر)،(۴)پیاده سازی یک سبک واحد برای همساز کردن ظاهر مستندو(۵)اعتبارسنجی بخش ها و فصا های برای مستندات بزرگ.هرکدام از این ویژگی هامستلزم پیاده سازی یک عملکرد فرعی جداگانه در نرم افزاراست>هرکدام از این عملکردهای فرعی را نیز میتوان تجزیه کرد.

۲۴-۴ فرایند

فعالیت های چار(فصل۲)که فرایند نرم افزار را مشخص می کنند،برای تمامی پروژه های نرم افزاری قابل استفاده اند.مساله،انتخاب مدل فرایندی برای نرم افزاری که قراراست توسط تیم پروژه مهندسی شود.

تیم شما باید تصمیم بگیرد که کدام مدل فرایندی برای(۱)مشتریان که محصول را درخواست کرده اند و افرادی که کار را انجام میدهند،(۲)خصوصیات خوده محصول و(۳)محیط پروژه ای که تیم نرم افزاری در آن کار میکند،از همه مناسب تر است.هنگامی که مدل فرایندی انتخاب شد،تیم باید براساس مجموعه فعالیت های چارچوبی فرایند،یک برنامه ریزی مقدماتی برای پروژه تعریف کند.یعنی،یک برنامه ریزی کامل که وظایف کاری لازم برای ایجاد فعالیت های چارچوبی را منعکس کند.این فعالیت ها را اختصار در بخش های بعدی مورد کاوش قرار خواهیم داد و در فصل۲۶با جزئیات بیشتری به آن ها خواهیم پرداخت.

۱-٤-۲٤ امتزاج محصول فرایند

برنامه ریزی برای پروژه با امتزاج محصول و فرایند آغاز میشود.هر عملکردی که قراراست تیم شما مهندسی کند،باید از یک مجموعه فعالیت های چارچوبی عبور کند که برای سازمان نرم افزاری شما تعریف شده اند.

فرض کنید که سازمان،فعالیت های چارچوبی کلی(ارتباطات،برنامه ریزی،مدل یازی،ساخت و استقرار)بحث شده در فصل دو را اتخاذ کرده است.اعضای تیمی که روی یک عملکرد محصول کار میکنند،هرکدام از این این فعالیت های چارچوبی را روی آن اعمال کنند.در اصل،ماتریسی مشابه شکل ۱-۲۴ایجاد میشود.

عملکرد های محصول اصلی(دراین شکل،عملکرد های نرم افزاز واژه پرداز ذکر شده دربخش قبل،نشان داده شده است)درطرف ستون سمت چپ فهرست شده اند. فعالیت های چارچوبی در سطر بالایی فهرست شده اند.وظایف کاری مهندسی نرم افزار(برای هر فعالیت چارچوبی)درسطر بعدی وارد میشوند.وظیفه مدیر پروژه(و سایر اعضای تیم)براورد منابع لازم برای هرسلول از ماتریس،تاریخ آغاز و پایان وظایف مربوط به هر سلول و محصولات کاری تولید شده در نتیجه هر وظیفه است.این فعالیت ها در فصل ۲۶مورد بحث قرار خواهیم گرفت.

۲-٤-۲٤ تجزیه ی فرایند

یک تیم نرم افزاری باید در گزینش بهترین مدل فرایندی برای پروژه و وظایف مهندسی نرم افزار و تشکیل دهنده آن مدل از انعطاف پذیری بسیار بالایی برخوردار باشد.یک پروژه نسبتا کوچک مشابه به آنچه که در گذشته انجام شده است،به بهترین نحو با استفاده از ترتیبی خطی قابل انجام است.اگر مهلت تحویل چنان تنگ باشد که عملکرد کامل را نتوان در مهلت مقرر تحویل داد،ممکن است راهبرد افزایشی،بهترین انتخاب باشد.بطور مشابه،پروژه هایی با خصوصیات دیگر(مثلا خواسته های نامعین،فناوری پیشرفته،مشتریان مشکل پسند،توان بالقوه در استفاده ی مجدد)به انتخاب سایر مدل های فرایندی می انجامد.

هنگامی که مدل فرایندی انتخاب شد، چارچوب فرآیند بر آن تطبیق داده میشود.درهرمورد،چارچوب کلی فرایند که بعداًمورد بحث قرار میگیرد،قابل استفاده خواهد بود.این چارچوب برای مدل های خطی،مدل های تکراری و افزایشی و مدل های تکاملی و حتی مدل های همروند یا مونتاژ مولفه ها جواب خواهد داد.چارچوب فرایند،ماهیتی ثابت دارد و به عنوان مبنایی برای تمامی کارهای انجام شده توسط سازمان نرم افزاری عمل میکند.

به هر حال،وظایف کاری واقعی متغیرند.تجزیه ی فرایند،هنگامی آغاز میشود که مدیر پروژه بپرسد:«این فعالیت چارچوبی را چگونه انجام میدهیم؟»برای مثال،یک پروژه ی نسبتاً ساده و کوچک ممکن است برای فعالیت ارتباطات به وظایف کاری زیر نیاز

داشته باشد:

۱.تهیه ی فهرستی از مسائل

۲.دیدار با ذی نفع ها برای پرداختن به این مسائل

۳.تهیه ی بیان حوزه

۴.مرور بیان حوزه از تمامی جهت ها

۵.اصلاح بیان حوزه بنا به نیاز

این رویداد ها که ممکن است در مدت زمانی کمتر از ۴۸ ساعت به وقوع بپیوندند،نشان گر تجزیه ی فرایندی هستند که برای پروژه های کوچک و نسبتاًساده مناسب هستند.اکنون پروژه ای پیچیده تر را در نظر بگیرید که حوزه ای گسترده تر را دربر میگیرد و تاثیر تجاری چشمگیری دارد.چنین پروژه ای ممکن است نیازمند وظایف کاری زیر برای ارتباطات باشد:

۱.مرور درخواست مشتری

۲.برنامه ریزی یک نشست رسمی و تسهیل شده با تمامی ذی نفع ها

۳.اجرای پژوهش برای مشخص کردن راهکار پیشنهادی و رویکردی موجود

۴.تهیه ی یک«سند کاری» و دستورالعملی برای نشست رسمی

۵.اجرای نشست

۶.توسعه ریز مشخصه ایی که ویژگی های داده ای،عملیات و رفتاری نرم افزار را منعکس کند.

به طریق دیگر،توسعه یuse case هایی که نرم افزار را از دیدگاه کاربر توصیف کنند.

۷.مرور هرکدام از ریزمشخصه ها یاuse caseها برای صحت،سازگاری و فقدان ابهام.

۸.مونتاژ این ریز مشخصه ها برای رسیدن به مستندی برای تعیین حوزه.

۹.مرور مستندات تعیین حوزه یا جمع آوری use case از تمامی جهت ها

۱۰.اصلاح مستندات تعیین حوزه یا use caseها بنا به نیاز.

هردو پروژه یک فعالیت چارچوبی اجرا میکنند که آن را ارتباطات می نامیم،ولی تیم پروژه ی نخست،نیمی از وظایف مهندسی نرم افزار تیم دوم را انجام میدهد.

۲۴-۵ پروژه

به منظور مدیریت یک پروژه نرم افزاری موفق، باید بدانید چه چیزهایی ممکن است به خطا برود به طوری که بتوانید ازمشکلات بپرهیزید.جان ربل[Ree99]در یک مقاله عالی درباره ی پروژه های نرم افزاری ده علامت تعریف میکند که نشان میدهد یک پروژه ی سیستم های اطلاعاتی در معرض خطر است:

۱.متخصصان نرم افزار،نیاز های مشتری را نمیفهمند.

۲.حوزه ی محصول به خوبی تعریف نشده است

۳.تغییرات بخوبی مدیریت نمیشود

۴.فناوری انتخاب شده تغییر میکند.

۵. نیازهای تجاری،تغییر میکند[یاخوب تعریف نشده اند]

۶.مهلت ها واقع بینانه نیستند.

۷.کاربران مقاوم هستند.

۸.حمایت از دست میرود.(یاهرگز بطور مناسب به دست نیامده است.)

۹.تیم پروژه فاقد افرادی با مهارت های مناسب است.

۱۰.مدیران(ودست اندرکاران)از بهترین کارها و درس های فراگرفته پرهیز میکنند.

حرفه ای های این صنعت ،هنگام بحث درباره ی پروژه های نرم افزاری دشوار،قالباً به قاعده ی ۹۰-۹۰۱ اشاره میکنند.نود درصد نخست یک سیستم،نود درصد از کل کار وزمان تخصیص یافته را میگیرد[Zah94]ریشه های قاعده ی ۹۰-۹۰را میتوان در نشانه های ذکر شده در فهرست بالا یافت اما بدبینی کافی است!یک مدیر چگونه باید عمل کند تا از مسائل ذکر شده در بالا بپرهیزد؟ریل[Ree99]یک رویکرد پنج بخشی مبتنی بر عقل سلیم برای پروژه های نرم افزاری پیشنهاد میکند.

۱.برای شروع،درست گام بردارید. برای این منظور باید سخت(خیلی سخت)کارکنید تا مساله ای را که قرار است حل شود،بفهمید و سپس اهداف و انتظارات واقع بینانه ای برای تمامی افراد درگیر در پروژه وضع کنید.تشکیل تیم درست (بخش۳-۲-۲۴)و اعطای خودمختاری،ابتکار عمل و فن آوری لازم به اعضای تیم، این وضعیت را تقویت میکنئ.

۲.نیروی پیشرانه را حفظ کنید بسیاری از پروژه ها شروع خوبی دارند ولی به آهستگی دچار از هم پاشیدگی میشوند.برای حفظ نیروی پیشرانه،مدیر پروژه باید انگیزه هایی ایجاد کند که رویگردانی اعضای تیم از مسیر اصلی را به حداقل برساند،تیم باید در هر وظیفه ای که به انجام میرساند،برکیفیت تاکید کند و مدیر ارشد باد هرکاری را که لازم است،انجام دهدتا سرراه تیم قرار نگیرد.

۳.فرایند را ردگیری کنید.برای یک پروژه نرم افزاری،پیشرفت به تولید محصولات کاری(مثلاًمدل ها، کدهای منبع،مجموعه ی موارد آزمون)به عنوان بخشی از فعالیت تضمین کیفیت(بااستفاده از مرور های فنی)تولید و تصویب میشوند.بعلاوه فرایند نرم افزاری و موازین پروژه(فصل۲۵)را میتوان جمع آوری کرد و برای ارزشیابی پیشرفت کار بر حسب میانگین های توسعه یافته برا سازمان توسعه ی نرم افزار بکاربرد

۴.هوش مندانه تصمیم بگیرید. در اصل،تصمبم هایی که مدیر پروژه و تیم نرم افزاری میگیرند،باید به گونه ای باشد که «سادگی»حفظ شود.هرگاه امکان داشت،تصمیم بگیرید که از نرم افزار های تجاری،یا مولفه ها و الگوهای موجود در بازار استفاده کنید،تصمیم بگیرید که در صورت دسترسی به روش های استاندارد از واسطه های سفارشی پرهیز کنید و تصمیم بگیرید به وظایف پیچیده یا خطرناک، زمانی بیش از آنکه فکر میکنید لازم باشد،اختصاص دهید(به هر دقیقه از این زمان نیاز خواهید داشت)

۵.پس از اتمام پروژه آن را کالبد شکافی کنید.برای استخراج درس هایی که از هرپروژه میگیرید،یک سازوکار مناسب وضع کنید.زمان بندی های برنامه ریزی شده و واقعی را ارزیابی کنید،معیارهای پروژه نرم افزاری را جمع آوری و تحلیل کنید،از مشتریان اعضای تیم،بازخوردبگیرید و یافته های خود را به شکل کتبی ثبت کنید.

۶-۲۴ اصل $W^2\ HH$

بری بوهم دریک مقاله ی عالی درباره ی پروژه ها و فرایند های نرم افزاری چنین میگوید:«شما به یک عصر سازماندهی نیاز دارید که به فراخور پروژه های ساده،طرح های ساده فراهم آورد.»بوهم،رویکردی را پیشنهاد میکند که به اهداف پروژه،نقاط عطف و زمانبندی ها،مسئولیت ها،رویکرد های فنی و مدیریتی و منابع مورد نیاز میپردازد.او این رویکرد را اصل $W^2\ HH$ مینامد.این رویکرد از پس یک سری پرسش تعیین میشود که پاسخ به آن ها به تعییت خصوصیات کلیدی پروژه و برنامه ریزی پروژه می انجامد:

این سیستم چرا تهیه میشود؟همه ذی نفع ها باید اعتبار دلایل تجاری کار نرم افزاری را مورد ارزیابی قرار دهند.آیا هدف تجاری،صرف هزینه و وقت آدم ها را توجیه میکند؟

چه کاری انجام خواهد شد؟مجموعه کارهای لازم برای پروژه تعریف میشود.

در چه زمانی لنجلم خواهد شد؟تیم با تعیین اینکه وظایف پروژه در چه زمانی باید انجام شود و چه زمانی باید به نقاط عطف رسید،زمانبندی پروژه را تعیین میکند.

چه کسی مسئول هر وظیفه ای است؟نقش هریک از اعضای تیم نرم افزاری تعیین میشود.

به لجاظ سازمانی چه جایگاهی دارند؟همه ی مسئولیت ها و نقش ها به دست اندرکاران نرم افزاری خلاصه نمیشود.

کار به لحاظ فنی و مدیریتی چگونه انجام خواهد شد؟هنگامی که حوزه محصول تعیین شد،یک راهبرد فنی و مدیریتی برای پروژه باید تعیین شود.

از هرکدام از منابع چه میزان مورد نیاز است؟پاسه این پرسش با برآورد های به عمل آمده براساس پاسخ پرسش های قبل بدست می آید.

اصل $W^2\ HH$ بوهم فارغ از اندازه یا پیچیدگی،برای هر پروژه نرم افزاری قابل استفاده است.پرسش های ذکر شده در بالا،خلاصه ای عالی از برنامه ریزی برای شما و تیم تان فراهم می آورند.

ابزارهای نرم افزاری

ابزار هاینرم افزاری برای مدیران پروژه

«ابزارهای»فهرست شده در اینجا جنبه ی عمومی دارندو برای گسترده ی وسیعی فعالیت های مدیر پروژه قابل استفلده اند.ابزار های تخصصی مدیریت پروژه(مانند ابزار های زمانبندی،ابزار های برابر،ابزار های تحلیل ریسک)در فصل های آینده معرفی خواهد شد.

ابزار های نمونه

Project Contorol یک ابزار ساده بنام (www.spmn.com) Softare Program Manager's Network Panelتهیه کرده است که وضعیت پروژه را بطور مستقیم به آگاهی مدیران پروژه میرساند.این ابزار یک سری «آمپر»دارد که مثل جلو پنجره ی خودروهاست وبا Microsoft Excelپیاده سازی میشود.آن را میتوانید از نشانی زیر دانلود کنید

www.spmn.com/prodicts-software.html

Ganthead.com (www.ganthead.com/) یک مجموعه چک لیست های مفید برای مدیران پروژه تهیه کرده است.

ittoolkit.com (**www.ittoolkit.com**)مجموعه ای از راهنماهای برنامه ریزی،الگوهای فرایند و کاربرگ های هوشمند»تهیه کرده است که روی CD-ROM قابل تهیه است.

۲۴-۷ اقدامات حیاتی

انجمن ایرلای فهرستی از «اقدامات حیاتی برای مدیریت مبتنی بر کارایی»تهیه کرده است.اینااقدامات بطور سازگار توسط پروژه های نرم افزاری بسیار موفقی استفاده میشوند و حیاتی به شمار میروند که کارایی رایج در آن ها به مراتب بهتر از میانگین های موجود در صنعت نرم افزار است.[Air99]

اقدامات حیاتی عبارت اند از: مدیریت پروژه بر اساس معیار ها(فصل۲۵)،برآورد تجربی هزینه و زمان بندی(فصل های ۲۶و۲۷)،پیگیری ارزش های بدست آمده(فصل۲۷)،ردگیری نقائص برحسب هدف های کیفیتی(فصل های ۱۴ تا ۱۶)و مدیریت منابع انسانی(بخش۲-۲۴).به هرکدام از این اقدامات حیاتی در بخش های ۳ و ۴ این کتاب پرداخته شده است.

پایان