

زمان بندی و ردگیری پروژه

فصل ۷

مفاهیم کلیدی (مرتب بر حروف الفبا)

ارزش بدست آمده ، افراد و نیروی کار ، تاخیر جدول زمان بندی ، ردگیری خطا ، ساختار تجزیه کار ، شبکه وظائف ، مجموعه وظائف ، مسیر بحرانی ، معیار تطابق ، نمودار خطی زمانی

KEY CONCEPTS

Adaptation criteria , critical pd , earned value , error tracking , lateness , people and effort , project plan , project tracking scheduling , principles , task network , task set , timeline chart work breakdown , structure

نگاه اجمالی

زمان بندی و ردگیری پروژه چیست؟ شما یک مدل فرآیند مناسب انتخاب کرده اید، شما کارهایی را که مهندس نرم افزار باید انجام دهد مشخص کرده اید، شما مقدار کار و تعداد افراد مورد نیاز را برآورد کرده اید، شما از مهلت مقرر آگاهید، و حتی ریسک ها و خطرهای را نیز مورد ملاحظه قرار داده اید. اکنون زمان وصل کردن نقطه ها به همدیگر است. یعنی شما باید یک شبکه از وظائف مهندسی نرم افزار بوجود آورید که شما را قادر خواهد کرد کارها را به موقع انجام دهید. زمانی که شبکه بوجود آید، شما باید مسئولیت تمام وظائف را محول کنید، البته بعد از اطمینان از انجام شدن کارها، و نیز باید شبکه را با بوقوع پیوستن خطرات تطبیق نمایید. به عبارت ساده تر، این زمان بندی و پیگیری طرح می باشد.

چه کسی آنها انجام می دهد؟ در سطوح طرح، مدیران طرح نرم افزار اطلاعاتی را که مهندسان نرم افزار ارائه می دهند بکار می برند و در یک سطح فردی مهندسان نرم افزار خود کار را انجام می دهند.

دلیل اهمیت آن چیست؟ بمنظور ایجاد یک سیستم پیچیده بسیاری از وظائف مهندسی نرم افزار به طور موازی انجام می گیرند، و نتیجه کاری که در یک مرحله انجام گرفته است ممکن است تأثیر عمیقی بر آنچه که در مرحله دیگر انجام می گیرد داشته باشد. درک این ارتباطات متقابل بدون داشتن یک برنامه زمان بندی بسیار مشکل می باشد. همچنین، تقریباً ارزیابی پیشرفت یک پروژه نرم افزاری بزرگ یا متوسط بدون داشتن یک برنامه زمان بندی جزئی غیرممکن است.

چه قدم هایی باید برداشته شود؟ وظائف مهندسی نرم افزار که با مدل فرآیند نرم افزاری تعیین می شود برای عملی کردن کار پالایش می گردند. نیروی کاری و مدت زمان لازم برای هر وظیفه در نظر

گرفته می‌شود و یک شبکه وظائف (که شبکه فعالیت نیز گفته می‌شود) ایجاد می‌گردد، بطوریکه تیم نرم‌افزاری را قادر می‌سازد در موعد مقرر کار را به انجام برساند.

حاصل کار چیست؟ برنامه زمانبندی پروژه و اطلاعات مربوط به آن حاصل خواهند شد.

چگونه مطمئن شوم که کار را درست انجام داده‌ام؟ زمانبندی مناسب مستلزم رعایت نکات ذیل

است:

۱- ظهور تمام وظائف در شبکه.

۲- اختصاص زمان و تلاش آگاهانه به هر وظیفه.

۳- نشان دادن مناسب ارتباط متقابل بین وظائف.

۴- اختصاص منابع به کارهایی که باید انجام گیرند.

۵- در نظر گرفتن بنیان‌های نزدیک به هم، بطوریکه پیشرفت کار قابل پی‌گیری باشد.

در اواخر دهه شصت قرن گذشته مهندس جوانی که چشمان روشنی داشت انتخاب گردید تا یک برنامه کامپیوتری برای یک کاربرد خودکار صنعتی بنویسد. دلیل انتخاب شدن او ساده بود، او تنها فرد در گروه فنی خود بود که در یک سمینار برنامه‌سازی کامپیوتری شرکت کرده بود. او ورود و خروج زبان اسمبلی و فرترن را می‌شناخت، اما چیزی درباره مهندسی نرم‌افزار نمی‌دانست و درباره زمانبندی و پیگیری طرح، اطلاعات او به مراتب کمتر نیز بود.

کارفرمایش کتابهای مناسب و نیز توصیفات شفاهی درباره آنچه او باید انجام می‌داد در اختیارش گذاشت. به او گفته شد که طرح مذکور باید در دو ماه تکمیل گردد.

او مبانی مربوط به کار خود را مطالعه کرد، رهیافت خود را مورد ملاحظه قرار داد، و شروع به نوشتن برنامه کرد. بعد از دو هفته کار فرمایش او را به دفتر کارش دعوت کرد و از چگونگی انجام کارها پرسید.

مهندس جوان با اشتیاق و سرزندگی پاسخ داد «خیلی خوب، درواقع خیلی راحت‌تر از آنچه من فکرش را می‌کردم. من تقریباً نزدیک به ۷۵ درصد کار را تمام کرده‌ام.»

کارفرما لبخند زد و گفت: «واقعاً عالی» و مهندس جوان را تشویق به انجام مطلوب کار کرد. آنها قرار گذاشتند که در طول یک هفته دوباره همدیگر را ملاقات کنند.

یک هفته بعد کارفرما مهندس را به دفترش فراخواند و پرسید: «ما در کجای کار هستیم؟»

مهندس جوان گفت: «همه چیز به خوبی در حال پیشرفت است، مشکلات کوچکی هم برای من

بوجود آمده است. ولی آنها را برطرف خواهم کرد و همه چیز روبراه خواهد شد.»

کارفرما پرسید: «ظرفتان راجع به موعد مقرر چیست؟»

مهندس جواب داد: «مشکلی نیست، من تقریباً ۹۰ درصد کار را به انجام رسانده‌ام.»

اگر شما بیش از چند سال در جهان نرم‌افزار کار کرده باشید، می‌توانید داستان را تمام کنید. تعجبی نخواهید داشت که مهندس جوان^۱ ۹۰ درصد کار را در زمان کلی انجام پروژه انجام داد و تنها پس از یک ماه تأخیر (با کمک دیگران) آنرا به پایان رسانید.

در خلال دهه گذشته این داستان دهها هزار بار توسط دست اندرکاران نرم‌افزار تکرار شده است.

سؤال اصلی این است، چرا؟

۷-۱ مفاهیم اولیه

اگر چه دلایل زیادی برای تأخیر در تحویل نرم‌افزار وجود دارد، اما بیشتر آنها را می‌توان به یکی یا بیشتر از علل اصلی زیر مربوط دانست:

- یک موعّد زمانی غیرقابل تحقق بوسیله فردی خارج از گروه توسعه نرم‌افزار مقرر گردید و بر مدیران و کارورزان داخل گروه فشار می‌آورد.
- نیازمندیهای درحال تغییر مشتری که در تغییرات برنامه زمانبندی منعکس نشده‌اند.
- کمتر از حد برآورد کردن ساده لوحانه میزان تلاش و / یا تعداد منابعی که برای انجام کار موردنیاز بودند.
- خطرات قابل پیش‌بینی و / یا غیرقابل پیش‌بینی که به هنگام شروع پروژه مورد ملاحظه قرار نگرفته بودند.

- مشکلات فنی که از قبل قابل پیش‌بینی نبوده‌اند.
- مشکلات انسانی که از قبل قابل پیش‌بینی نبوده‌اند.
- ارتباط غلط بین اجرا کنندگان پروژه که تأخیرها را در پی داشته است.
- عدم توانایی مدیریت پروژه در تشخیص انجام نشدن پروژه در برنامه زمانبندی انجام شده و فقدان عمل برای برطرف کردن مشکل.

موعدهای غیرقابل تحقق، حقایقی از زندگی در تجارت نرم‌افزار می‌باشند. بعضی وقتها از نظر فردی که موعّد را مقرر کرده است، این موعدها به دلایلی مشروع مقرر می‌گردند، اما عقل سلیم می‌گوید که مشروعیت را باید کسانی که کار را انجام می‌دهند تعیین کنند.

۷-۱-۱ توضیحی بر "دیرکرد"

زمانی ناپلئون گفته بود: «هر فرمانده ارشدی که اقدام به انجام طرحی می‌کند که او شکست آنرا نشانه‌ای از کوتاهی می‌داند، باید دلایلش را بیان کند، از پافشاری بر روی طرح خودداری کند و سرانجام

نقل قول

زمان بندی غیر معقول
یا مفرط محتمل است
که بیشترین اثر
تخریبی را برای کل
نرم افزار داشته باشد.
کاپرز جونز

نقل قول

من تعیین زمان تحویل را
عاشقانه دوست دارم.
از شنیدن صدای صغیر بالها
که نشانگر لحظه پرواز است.
دلخوشم.
داگلاس آدامز

۱. اگر شما مشتاقانه گمان برده اید که این یکی از خاطرات نویسنده (پرسمن) است، درست حدس زده اید!

بجای اینکه ابزار اضمحلال ارزش او باشد استعفايش را تقدیم نماید.» گفته‌های مؤثری که هر مدیر پروژه نرم‌افزاری باید درباره آنها تعمق کند.

برآورد و فعالیتهای تجزیه و تحلیل خطر در فصل‌های ۵ و ۶ بحث شدند، و فنون زمانبندی‌ای که در این فصل توصیف شدند اغلب تحت تأثیر یک موعد تعریف شده انجام می‌شوند. اگر بهترین ارزیابی‌ها نشان می‌دهند که موعد مقرر غیرقابل تحقق است، یک مدیر پروژه قابل باید «گروه خود را در برابر فشارهای (زمانبندی) بی دلیل حمایت کند ... [و] فشارها را به مسببین آنها برگرداند.» [PAG85]^۱

بمنظور روشن شدن مسئله، فرض کنید که لایک گروه توسعه نرم‌افزاری خواسته شده است که یک کنترل کننده زمان حقیقی برای یک وسیله تشخیص پزشکی بسازند که قرار است در طول نه ماه وارد بازار گردد. بعد از ارزیابی دقیق و تجزیه و تحلیل خطر، مدیر پروژه نرم‌افزاری به این نتیجه می‌رسد که نرم‌افزار مورد نظر برای ساخت با پرسنل موجود، نیاز به یک زمان ۱۴ ماهه دارد. اکنون مدیر پروژه باید چه کاری بکند؟



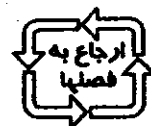
هنگامی که مدیریت محدودیت زمانی غیرممکنی را درخواست می‌کند، چه باید کرد؟

رفتن به دفتر کار مشتری (در این مورد مشتری احتمالی بازاریابی/فروش می‌باشد) و تقاضای تغییر زمان تحویل غیرواقعی بنظر می‌رسد. فشار بازار بیرون آن تاریخ را دیکته کرده است و کلاً باید وارد بازار گردد. از طرف دیگر امتناع از انجام کار نیز به همان میزان احمقانه است (از نظر سابقه شغلی). بنابراین، چه باید کرد؟

در این موقعیت مراحل ذیل توصیه می‌گردند:

- ۱- با بکار بردن داده‌های پروژه‌های قبلی دست به یک برآورد جزئی بزنید. تلاش‌ها و زمان برآورده شده را برای پروژه تعیین کنید.
- ۲- با بکار بردن یک مدل فرایند افزایشی (فصل ۲)، راهبرد مهندسی نرم‌افزاری بوجود آورید که عملکرد بحرانی ناشی از موعد مقرر را برطرف خواهد کرد، اما عملکرد دیگر را تا زمانی بعد را به تأخیر بیندازید. طرح را مستند کنید.

- ۳- با مشتری ملاقات کنید (با بکار بردن برآورد جزئی)، برای او توضیح دهید که چرا موعد مقرر غیرقابل تحقق است. با اطمینان اشاره کنید که همه برآوردها بر مبنای عملکرد پروژه‌های قبلی انجام گرفته است. همچنین با اطمینان درصد بهبودی را که برای محقق کردن موعد مقرر کنونی موردنیاز خواهد بود را نشان دهید.^۲ توصیه زیرا کارآمد می‌باشد:



مدل های افزایشی فرایند، در فصل ۲ توضیح داده شده اند.

«من فکر می‌کنم که ما ممکن است با تاریخ تحویل کنترل کننده XYZ دارای مشکل باشیم. من جزئیات خلاصه شده نرخ های تولید پروژه‌های قبلی و نیز برآوردی که ما از راههای متفاوت دیگری

1. Page-Jones, M.

۲. اگر پیشرفت ۱۰ تا ۲۵ درصد باشد، ممکن است که کار انجام شود، اما اکثراً گمان دارند که باید بیش از ۵۰ درصد پیشرفت در عملکرد تیم وجود داشته باشد. این یک نفع غیرواقعی است.

کرده‌ایم در اختیار شما قرار داده‌ام. شما در خواهید یافت که من یک بهبود ۲۰ درصدی در نرخ‌های تولید قبلی فرض کرده‌ام، با این حال ما همچنان نیاز به یک تاریخ تحویل ۱۴ ماهه بجای یک مدت ۹ ماهه داریم.»

۴- راهبرد توسعه افزایشی را به عنوان جایگزینی ارائه دهید.

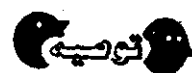
«ما چند انتخاب داریم، و من امیدوارم شما بر مبنای آنها تصمیمی اتخاذ کنید. اولین انتخاب این است که ما افزایشی در بودجه داشته باشیم و منابعی دیگری را بکار اندازیم بطوریکه ما بتوانیم با سرعتی بالا این کار را در نه ماه انجام دهیم. اما در نظر داشته باشید که پدیل فشردگی مدت زمانی خطر پایین آمدن کیفیت وجود دارد.^۱ دوم اینکه ما می‌توانیم بعضی از کاردکرها و قابلیت‌های نرم‌افزار مورد تقاضای شما را حذف نماییم، که این نیز از کارایی نوع اولیه کالا تا اندازه‌ای می‌کاهد. اما می‌توانیم تمام کارایی‌های کالا را تهیه نماییم و سپس بعد از یک دوره ۱۴ ماهه آنرا تحویل دهیم. سومین انتخاب نادیده گرفتن واقعیت و امیدواری برای انجام پروژه در ۹ ماه می‌باشد. در این صورت ما بی‌دلیل عرصه را تنگ کرده‌ایم که می‌تواند به از دست دادن مشتری منجر گردد. امیدوارم که شما با من موافقت نمایید که انتخاب سوم غیرقابل قبول است. پیشینه گذشته و بهترین ارزیابی‌های ما می‌گوید که آن انتخاب غیرواقعی است و ما را با شکست مواجه می‌کند.»

البته اعتراضاتی وجود خواهد داشت، اگر ارزیابی‌هایی مطمئن بر مبنای اطلاعات تاریخی خوب ارائه گردد، احتمالاً انتخاب‌های بحث شده ۱ یا ۲ برگزیده خواهند شد [و بدین ترتیب] موعد مقرر غیرواقعی قبلی ناپدید می‌شود.

۷-۱-۲ اصول پایه

زمانی از فرد بروکس، نویسنده معروف کتاب مرد افسانه‌ای ماه^۲ [BRO95] پرسیده شد چگونه پروژه‌های نرم‌افزاری از برنامه زمان‌بندی عقب می‌افتند. پاسخ او ساده و عمیق بود: «یک وقتی در یک روزی».

واقعیت یک پروژه فنی (چه ساخت یک دستگاه هیدروالکترونیک باشد و چه توسعه یک سیستم عامل) انجام صدها وظیفه کوچکتر برای کامل شدن یک هدف بزرگتر است. بعضی از این وظایف خارج از جریان اصلی قرار دارند و ممکن است بدون نگرانی از تأثیری که بر زمان تکمیل پروژه می‌توانند داشته باشند کامل



کارهایی که برای اهداف مدیریت پروژه باید انجام گیرد، نیازی نیست که دستی صورت پذیرد، ابزارهای زمان‌بندی ممتازی برای پروژه‌ها وجود دارند، آنها را به کار برید.

۱. شما ممکن است کاه افراد بیشتری را دخیل در کار کنید اما تقویم زمانی کاهش نیابد.

2. Man-Month

3. Brooks, M.

گردد. بعضی کارهای دیگر بر روی «مسیر بحرانی» قرار می‌گیرند.^۱ اگر انجام این کارهای «بحرانی» به تعویق افتد، تاریخ اتمام کل پروژه در مخاطره قرار می‌گیرد.

هدف مدیر پروژه توصیف همه کارهای پروژه، ساختن شبکه‌ای که روابط متقابل آنها را شرح می‌دهد، شناسایی کارهایی که در شبکه بحرانی می‌باشند، و بعد پی‌گیری پیشرفت آنها برای اطمینان از تشخیص تأخیر «یک وقتی در یک روزی» می‌باشد. برای تکمیل شدن کار، مدیر باید دارای برنامه زمان‌بندی باشد که تا حدی دارای ثبات در تعریف باشد بطوریکه مدیر را قادر سازد پیشرفت پروژه را چک کند و آنرا کنترل نماید.

زمان‌بندی پروژه نرم‌افزاری^۲ فعالیتی است که نیروی انسانی برآورد شده را در مدت برنامه‌ریزی شده پروژه از راه توزیع نیروی انسانی نسبت به کارهای مهندسی نرم‌افزاری خاص، تنظیم می‌کند. با این حال، ذکر این نکته حائز اهمیت است که برنامه زمان‌بندی با گذشت زمان کامل می‌شود. در خلال مراحل اولیه برنامه‌ریزی طرح، یک زمان‌بندی ماکروسکوپی^۳ بوجود آمده است. این نوع زمان‌بندی تمام فعالیت‌های مهندسی نرم‌افزاری عمده را شناسایی می‌کند و کارکردهایی را تولید می‌کند که بکار برده می‌شوند. با جلو رفتن پروژه‌ها هر مدخل در زمان‌بندی ماکروسکوپی به یک برنامه زمان‌بندی جزئی^۴ تبدیل می‌گردد. در اینجا، کارهای نرم‌افزاری خاص (برای انجام یک فعالیت موردنیازند) شناسایی و زمان‌بندی شده‌اند.

زمان‌بندی برای پروژه‌های مهندسی نرم‌افزاری از دو دیدگاه نسبتاً متفاوت می‌تواند مورد بررسی قرار گیرد. در دیدگاه نخست، یک زمان پایانی برای عرضه یک سیستم مبتنی بر کامپیوتر از قبل (که غیرقابل بازگشت) برپا شده است. سازمان نرم‌افزاری مجبور است که تلاش خود را در قالب زمان تجویز شده توزیع نماید. دیدگاه دوم زمان‌بندی نرم‌افزاری فرض می‌دارد که قیود زمانی سختگیرانه مورد بحث

واقع شده است اما زمان پایان توسط سازمان مهندسی نرم‌افزاری تنظیم شده است. نیروی کاری به نحوی توزیع شده‌اند تا از منابع به بهترین نحو استفاده گردد و هر زمان پایانی بعد از تجزیه و تحلیل دقیق نرم‌افزار تعریف گردیده است. متأسفانه، بسیار بیشتر از موقعیت دوم با موقعیت نخست برخورد می‌شود. مانند تمام زمینه‌های دیگر مهندسی نرم‌افزار، تعدادی اصول اساسی، زمان‌بندی پروژه نرم‌افزاری را جهت می‌دهند:

سواسازی. پروژه باید به تعدادی فعالیت‌ها و وظائف قابل مدیریت تقسیم گردد. برای به انجام رساندن این کار، هم تولید و هم روند کار تجزیه می‌شوند. (فصل ۳)

نقل قول

زمان بندی خوش
بینانه نتیجه ای بهتر از
زمان بندی واقعی
نخواهد داشت، می
توان گفت که طولانی
تر از آن هم خواهد
شد. استیو مک کانل



هنگامی که شما یک
برنامه زمان بندی را
تنظیم می کنید، کار را
به قسمتهای مجزا
تفکیک کرده، وابستگی
های داخلی کارها را
مشخص سازید، نیرو و
زمان هر کار را معین
کنید، مسئولیت هر کار
را معلوم کنید و نقاط
بررسی و مراحل را
مشخص نمایید.

۱. مسیر بحرانی در ادامه بحث به تفصیل توضیح داده خواهد شد.

2. software project scheduling

3. macroscopic schedule

4. detailed schedule

وابستگی متقابل (داخلی). وابستگی‌های متقابل هر فعالیت یا وظیفه سوا شده‌ای باید تعیین گردد. بعضی کارها باید به صورت متوالی انجام گیرد درحالی‌که بعضی دیگر به صورت موازی انجام می‌گیرد. بعضی فعالیت‌ها تا زمانی که تولید کار دیگری در دسترس نباشد قابل آغاز شدن نیستند. فعالیت‌های دیگری می‌توانند به طور مستقل انجام گیرند.

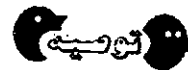
تخصیص زمان. برای زمان‌بندی هر کاری تعدادی واحد کاری باید به آن اختصاص داده شود (مثلاً، نفر-روزهای نیروی کاری). همچنین برای هروظیفه باید یک تاریخ شروع و یک تاریخ اتمام مشخص شود که یکی از کارکردهای وابستگی‌های متقابل است و باید مشخص شود که آیا اساس انجام کار پاره وقت است یا تمام وقت.

اعتبارسنجی نیروی کاری. هر پروژه‌ای دارای تعداد معینی پرسنل می‌باشد. به هنگام تخصیص زمان، مدیر پروژه باید اطمینان حاصل کند که در هیچ زمانی تعدادی بیشتر از افراد در نظر گرفته شده در برنامه کاری نباشند. بعنوان مثال، پروژه‌ای را در نظر بگیرید که پرسنل اختصاص داده شده به آن، سه نفر می‌باشند. (مثلاً ۳ نفر-روز در هر روز نیروی کاری اختصاص داده شده موجود می‌باشد). در یک روز خاص، ۷ کار همزمان باید انجام گیرد. هر وظیفه ۰/۵ روز تلاش فردی نیاز دارد. در اینجا تلاش بیشتر از آنچه افراد می‌توانند انجام دهند اختصاص داده شده است.

مسئولیت‌های معین. هر کار زمان‌بندی شده‌ای باید به عضو یک گروه خاص اختصاص داد. نتایج معین. هر کار زمان‌بندی شده‌ای باید یک نتیجه معین داشته باشد. برای پروژه‌های نرم‌افزاری، نتیجه معمولاً یک محصول کاری (مثلاً، طراحی یک پیمانه) یا بخشی از یک محصول کاری است. محصولات کاری اغلب به صورت (محصولات) قابل عرضه با هم ترکیب می‌شوند.

مراحل برجسته معین. هر وظیفه یا مجموعه باید در کنار یک مرحله برجسته پروژه قرار گیرد. یک مرحله برجسته وقتی حاصل می‌گردد که یک یا چند تولید کاری از نظر کیفیت مورد مطالعه قرار گرفته‌اند (فصل ۸) و مورد قبول واقع شده‌اند.

هر یک از اصول فوق همزمان با تکمیل زمان‌بندی پروژه بکار بسته می‌شوند.



اگر قرار است که افراد

بیشتری را در پروژه به

تاخیر افتاده ای، فعال

کنید دقت کنید که

تقسیم وظایف روشن و

گویا باشد (وگرنه گرفتاری

های بیشتری خواهید

داشت)

۷-۲ رابطه میان افراد و نیروی کار

در یک پروژه توسعه نرم‌افزاری کوچک یک فرد به تنهایی می‌تواند نیازمندیها را تجزیه و تحلیل کند، طرح اجرا کند، برنامه بنویسد، و آزمون‌ها را اجرا کند. با افزایش اندازه یک پروژه، افراد بیشتری نیز باید در آن درگیر شوند. (ما بندرت می‌توانیم تلاش فردی را تصور کنیم که یک کار ده نفر-سال را، طی ده سال انجام دهد!)

افسانه معروفی وجود دارد (که در فصل ۱ مورد بحث واقع شد) که هنوز هم توسط بسیاری از مدیرانی که مسئول توسعه نرم‌افزاری هستند باور می‌شود: «اگر ما از برنامه عقب باشیم، با افزودن برنامه‌نویسان بیشتر، تأخیر را جبران خواهیم کرد.» متأسفانه، افزودن افراد دیگری به پروژه، تأثیر مخربی بر پروژه خواهد داشت و حتی باعث می‌شود که برنامه زمانبندی بیشتر به هم بخورد. افرادی که به سیستم اضافه می‌شوند باید با سیستم آشنا گردند، و آنها که مسئول آموزش آنها می‌شوند همانهایی هستند که کار را انجام می‌دهند. وقتی که در حال آموزش دیگران هستند، کاری انجام نمی‌دهند و بنابراین پروژه بیشتر عقب می‌افتد.

افزون بر زمانی که برای آشنایی با سیستم صرف می‌شود، افراد بیشتر باعث بالا رفتن تعداد مسیرهای ارتباطی و در نتیجه پیچیدگی ارتباط در یک پروژه می‌شود. اگر چه ارتباط در موفقیت توسعه نرم‌افزاری [نقشی] کاملاً اساسی دارد، اما هر مسیر ارتباطی جدیدی، نیاز به تلاش افزون‌تر و در نتیجه زمان بیشتری دارد.

۷-۲-۱ یک مثال

چهار مهندس نرم‌افزار را در نظر بگیرید، که هر یک به هنگام کار بر روی یک پروژه بطور مستقل قادرند ۵۰۰۰ خط برنامه در سال تولید کنند. وقتی این مهندسان در یک پروژه گروهی قرار می‌گیرند، ایجاد شش مسیر ارتباطی بالقوه ممکن می‌شود. هر مسیر ارتباطی نیاز به زمان دارد که در غیر این صورت صرف توسعه نرم‌افزار می‌شد. می‌توانیم فرض کنیم که قدرت تولید گروه (وقتی با LOC یا تعداد خطوط سنجیده می‌شود) برای هر مسیر ارتباطی به ۲۵۰ خط در سال کاهش خواهد یافت و این به سبب تعداد زیاد مسیرهای ارتباطی است. بنابراین قدرت تولید گروه برابر است با:

$$18180 = (250 \times 6) - 20000 \quad \text{خط در سال که ۷۵ درصد کمتر از آنچه‌ی است که ما باید انتظار داشته باشیم.}^2$$

پروژه یک ساله‌ای که تیم فوق بر روی آن در حال کار است از برنامه زمان‌بندی عقب می‌افتد، و دو ماه قبل از اتمام زمان انجام پروژه دو نفر دیگر به گروه اضافه می‌گردد. تعداد مسیرهای ارتباطی به ۱۴ افزایش می‌یابد. بهره‌وری ورودی پرسنل جدید برابر است با: $1680 = 840 \times 2$ خط در سال برای دو ماه باقی مانده قبل از عرضه محصول. بهره‌وری گروه اکنون برابر است با:

$$18180 = (250 \times 14) - 20000 + 1680 \quad \text{خط در سال}$$



رابطه میان تعداد
افرادی که در یک
پروژه نرم‌افزاری
مشغول می‌باشند و
بهره‌وری، خطی نمی‌باشد.

۱. در واقع، کمتر از سه نفر - روز خواهد بود. به دلایل نشست‌های غیر ضروری، بیماری‌ها، تعطیلات و بسیاری دلایل دیگر. با این وجود ما ۱۰۰ درصد دسترسی را برای این منظور لحاظ می‌کنیم.

۲. ممکن است که یک بحث متقابل مطرح شود: ارتباطات، که البته اگر موثر باشد، می‌تواند کیفیت کار را بهبود بخشد، به دنبال آن کاهش میزان کار مجدد و افزایش بهره‌وری شخصی اعضاء تیم را به دنبال داشته باشد. داوری پایان یافت.

اگرچه مثال فوق یک ساده سازی بی‌رویه کلی از وضعیت های جهان واقعی است، اما برای روشن شدن نکته کلیدی دیگری بکار می‌رود: رابطه بین تعداد افرادی که بر روی یک پروژه نرم‌افزاری کار می‌کنند و بهره‌وری کلی، خطی نمی‌باشد.

۷-۲-۲ یک رابطه تجربی

با به یادآوردن «معادله نرم‌افزار» [PUT92]^۱ که در فصل ۵ معرفی گردید، می‌توانیم رابطه بسیار غیرخطی بین مدت زمان تکمیل یک پروژه و تلاش انسانی و نیروی کار لازم برای تکمیل پروژه را نشان دهیم. تعداد خطوط مربوط به برنامه (جملات برنامه منبع)، L وابسته به تلاش و زمان توسعه در معادله زیر می‌باشد:

$$L = P \times E^{1/3} t^{4/3}$$

که E نیروی مورد نیاز برای توسعه بر حسب نفر - ماه می‌باشد، P پارامتر بهره‌وری است که گستره ای از عوامل مربوط به کار مهندسی نرم‌افزار با کیفیت بالا را نشان می‌دهد (توجاً مقدار P بین ۲۰۰۰ و ۱۲۰۰۰ خواهد بود)، و t زمان پروژه بر حسب ماه است.

ارایش مجدد این معادله ما را به معادله ای برای محاسبه E رهنمون می‌سازد:

$$E = L^3 / (P^3 t^4)$$

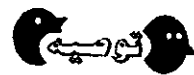
در این جا E تلاش و نیروی کاری مورد نیاز است (بر حسب نفر - سال) که بر چرخه کامل حیات، توسعه و نگهداری نرم‌افزار بسط داده شده است، و t زمان توسعه در سال‌ها می‌باشد. معادله توسعه تلاش را می‌توان با گنجاندن فاکتور نرخ سنگینی کار به هزینه توسعه مربوط ساخت. (\$/نفرسال).

چنین چیزی دارای نتایج جالبی خواهد بود. یک پروژه نرم‌افزار زمان واقعی را در نظر بگیرید که LOC یا تعداد خطوط برنامه ۳۳/۰۰۰ و ۱۲ سال تلاش فردی برآورد شده است. اگر ۸ نفر به تیم پروژه

اختصاص داده شوند، پروژه تقریباً در ۱/۳ سال تکمیل خواهد شد. با این حال، اگر زمان پایان پروژه را به ۱/۷۵ سال افزایش دهیم، ماهیت بسیار غیرخطی مدل که در معادله (۷-۱) توصیف شد نمایانده می‌شود:

$$E = L^3 / (P^3 t^4) \sim 3.8 \text{ نفر - سال}$$

از این معادله چنین برمی‌آید که با یک افزایش شش ماهه در زمان اتمام پروژه، می‌توانیم تعداد افراد را از ۸ به ۴ نفر کاهش دهیم! در مورد اعتبار این نتیجه‌گیری می‌توان بحث کرد، اما آنچه از آن برمی‌آید روشن است:



هرچه قدر زمان های

تحويل و تکمیل را

محدودتر سازید، به این

نکته می‌رسیم که کار

طبق برنامه زمانی

پیش نمی‌رود.

صرفنظر از نیروی

کاری که به آن

تخصیص داده اید، با

حقیقت، برخورداری

واقعی داشته باشید و

زمان بندی را واقع

بینانه و از نو تعریف

کنید.

به کارگیری تعداد کمتری از افراد در یک مدت کمی طولانی‌تر برای رسیدن به هدفی یکسان سودآور خواهد بود.

۷-۲-۳ توزیع نیروی کار

هر یک از فنون ارزیابی پروژه نرم‌افزاری که در فصل ۵ مورد بحث واقع شد منجر به ارزیابی واحدهای کار می‌شود (به عنوان مثال نفر-ماهها) که برای تکمیل توسعه نرم‌افزاری مورد نیازند. توزیع تلاشی که معمولاً در مراحل تعریف و توسعه به آن اشاره می‌شود قانون ۴۰-۲۰-۴۰ است.^۲ ۴۰ درصد تمام تلاش به طرح و تجزیه تحلیل ابتدایی و پایانی اختصاص داده شده است. و درصد مشابهی نیز به آزمون نهایی مربوط شده است. شما با کمی بررسی متوجه می‌شوید که به برنامه‌نویسی بی‌توجهی شده است (۲۰ درصد تلاش). از توزیع تلاش تنها باید به عنوان یک رهنمون استفاده کرد. ویژگی‌های هر پروژه خود باید توزیع تلاش را دیکته کند. تلاشی که مصروف برنامه‌ریزی پروژه می‌شود به‌ندرت بیش از ۲ یا ۳ درصد کل تلاش می‌باشد، مگر این‌که طرح دارای مخارج عظیمی برای یک سازمان باشد و خطر آن بسیار باشد. تجربه و تحلیل نیازمندیها ممکن است ۱۰ تا ۲۵ درصد تلاش پروژه را تشکیل دهد. تلاشی که صرف تجزیه و تحلیل یا الگوی‌سازی اولیه می‌شود باید نسبت به اندازه و پیچیدگی پروژه به‌طور مستقیم افزایش یابد. معمولاً تلاشی بین ۲۰ تا ۲۵ درصد به طرح نرم‌افزار مینول می‌گردد. زمانی که صرف مرور طرح و تکرارهای بعدی می‌شود نیز باید مورد ملاحظه قرار گیرد.

به سبب تلاشی که صرف طرح نرم‌افزار می‌شود، برنامه‌نویسی باید با مشکل نسبتاً کمتری انجام گیرد. و ۱۵ تا ۲۰ درصد از تلاش کلی را نیاز داشته باشد. اجرای آزمون و رفع اشکالات می‌تواند ۳۰ تا ۴۰ درصد از تلاش توسعه نرم‌افزار را به خود اختصاص دهد. حساس بودن نرم‌افزار اغلب تعیین‌کننده میزان آزمون مورد نیاز است. اگر نرم‌افزار به ارزیابی انسانی وابسته باشد (خرابی نرم‌افزاری ممکن است باعث کم شدن عمر [سیستم] گردد)، ممکن است حتی درصد بیشتری از تلاش مورد نیاز باشد.



چه میزان نیروی انسانی
باید برای هر فعالیت مهم
مهندسی نرم‌افزار پیش
بینی شود؟

۷-۳ تعریف یک مجموعه وظائف برای پروژه نرم‌افزاری

تعدادی از مدل‌های فرایند مختلف در فصل (۲) توصیف شدند. این مدل‌ها انگاره‌های مختلفی برای توسعه نرم‌افزار ارائه می‌دهند. صرف‌نظر از این‌که آیا یک تیم نرم‌افزاری یک الگوی متوالی خطی، یک

1.40-20-40 rule

۲. امروزه در پروژه‌های بزرگ نرم‌افزاری توصیه می‌شود که بیش از ۴۰ درصد تمام نیروی کار بر تحلیل و طراحی وظائف صرف شود. بنابراین نام ۴۰-۲۰-۴۰ شاید حس مشترکی را بر نیانگیزد.

الگوی تکرارکننده، یک الگوی تکاملی، یک الگوی همزمان یا جایگشت (Permutation) را انتخاب می‌کند یا نه، مدل فرآیند در بردارنده یک مجموعه وظایفی است که تیم نرم‌افزاری را قادر به تعریف، توسعه و نهایتاً پشتیبانی نرم‌افزار کامپیوتر می‌سازد.

مجموعه‌ای از وظایف که برای همه پروژه‌ها مناسب باشد وجود ندارد. مجموعه کاری که برای یک سیستم بزرگ و پیچیده مناسب است برای تولید نرم‌افزاری ساده و نسبتاً کوچکی احتمالاً بسیار افراطی خواهد بود. بنابراین یک فرآیند نرم‌افزاری مؤثر مجموعه‌ای از وظایف را تعریف خواهد کرد، که هر یک برای برآورده کردن نیازهای انواع گوناگونی از پروژه‌ها به کار می‌روند.

یک مجموعه وظایف^۱ مجموعه‌ای از کارهای مهندسی نرم‌افزار، هدف‌های اساسی و قابلیت‌های عرضه است که برای تکمیل یک پروژه خاص باید گردهم آیند. مجموعه وظایفی که انتخاب می‌گردد باید دارای نظم و ترتیب کافی برای رسیدن به کیفیت نرم‌افزاری بالا باشد. اما از طرف دیگر نباید تیم پروژه را متحمل کار غیرضروری نماید.

مجموعه‌های وظایف برای انطباق با انواع گوناگون پروژه‌ها و درجات استقامت مختلف طراحی شده‌اند. اگر چه بدست دادن یک طبقه‌بندی جامع از انواع پروژه‌های نرم‌افزاری مشکل است، بیشتر سازمان‌های نرم‌افزاری با پروژه‌های ذیل سر و کار دارند:

۱- پروژه‌های توسعه مفهوم^۲ که برای کشف بعضی مفاهیم تجاری جدید یا کاربرد بعضی فناوری‌های جدید آغاز گردیده‌اند.

۲- پروژه‌های توسعه برنامه کاربردی جدید^۳ که در نتیجه تقاضای یک مشتری خاص به آنها پرداخته شده است.

۳- پروژه‌های افزایش برنامه کاربردی^۴ که موقعی اجراء می‌می‌گردند که نرم‌افزار موجود متحمل اصلاحات بزرگی در رابطه با کارکرد، عملکرد یا رابطی که برای کاربر پایانی قابل مشاهده است، می‌شوند.

۴- پروژه‌های نگهداری برنامه کاربردی^۵ که به تصحیح، تطبیق یا گسترش نرم‌افزار موجود از راه‌هایی که ممکن است سریعاً برای کاربر قابل مشاهده نباشند می‌پردازند.

۵- پروژه‌های مهندسی مجدد^۶ که به منظور تجدید ساخت کلی یا جزیی سیستم موجود بدان‌ها پرداخته می‌شود.



یک "مجموعه وظایف"
مجموعه‌ای از کارهای
مهندسی نرم‌افزار است،
به همراه مراحل کار و
آنچه باید تحویل داده
شود.

1. task set

2. Concept development projects

3. New application development projects

4. Application enhancement projects

5. Application maintenance projects

6. Reengineering projects

حتی در درون یک نوع پروژه هم، عوامل زیادی وجود دارند که در انتخاب مجموعه وظایف مؤثر واقع می‌گردند. وقتی این عوامل با هم تلفیق می‌شوند، نشانه‌ای از میزان دقتی که با آن فرآیند نرم‌افزاری باید به کار بسته شود بدست می‌دهند.

۷-۳-۱ میزان سختی و دقت^۱

حتی در مورد پروژه‌ای از نوع خاص، میزان دقتی که با آن فرآیند نرم‌افزاری به کار بسته می‌شود به‌طور قابل ملاحظه‌ای تغییر می‌کند. میزان دقت، تابعی از ویژگی‌های بسیاری از پروژه‌هاست. به‌عنوان یک مثال، پروژه‌های حساس غیر تجاری کوچک را معمولاً می‌توان با دقت کمتری نسبت به کاربردهای حساس تجاری پیچیده بزرگ مورد ملاحظه قرار داد. با این وجود، لازم به ذکر است که تمام پروژه‌ها باید به‌گونه‌ای اجرا گردند که نتیجه آنها محصولات با کیفیت عالی و به هنگام باشد.

چهار میزان دقت متفاوت قابل تعریف می‌باشد:

معمولی. تمام فعالیت‌های چارچوب فرآیند اجرا شده‌اند (فصل ۲) اما تنها یک مجموعه وظایف حداقل مورد نیاز است. معمولاً، کارهای پوششی به حداقل خواهند رسید و نیازهای تبتی کاهش خواهند یافت. تمام قواعد اساسی مهندسی نرم‌افزار هنوز قابل کاربرد هستند.

ساختیافته. چارچوب فرآیند برای این پروژه به اجرا درخواهد آمد. فعالیت‌های شبکه و فعالیت‌های چتری و کارهای مربوطه متناسب با نوع پروژه به اجرا درخواهند آمد و فعالیت‌های چتری مورد نیاز برای تضمین کیفیت بالای کار به اجرا درخواهند آمد.

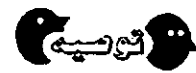
ثابت. تمام فرآیند در مورد این پروژه به اجرا درخواهد آمد، البته با درجه خاصی از نظم و ترتیب که موجب تضمین کیفیت بالای کار می‌شود. تمام فعالیت‌های چتری به اجرا درخواهند آمد و مستندسازی گسترده انجام خواهد گرفت.

واکنش سریع. شبکه فرآیند برای این پروژه به اجرا درخواهد آمد، اما به دلیل یک موقعیت اضطراری^۲ تنها آن کارهایی که برای رسیدن به کیفیت مطلوب اساسی‌اند به اجرا درخواهند آمد. ضبط - برگشتی

(مثلاً بدست دادن مجموعه کاملی از مستندسازی و انجام بازبینی‌های اضافی) بعد از این که کاربرد/محصول به مشتری تحویل گردید، انجام خواهد گرفت.



هر قدر انعطاف پذیری کاهش یابد و دقت زیاد شود، پیچیدگی وظایف و اندازه کارها رشد خواهد نمود.



اگر همه چیز را فوق العاده و ضروری انگارید، فرآیند نرم‌افزاری یا افرادی که مدیریت اداری را عهده دار هستند یا هر دو، در مواردی با مشکل مواجه خواهند شد.

1. degree of rigor

۲. وضعیت اضطراری به ندرت روی خواهد داد. (شاید در بافت مهندسی نرم‌افزار بیش از ۱۰ درصد کل موارد نباشد). یک وضعیت اضطراری متفاوت است با قیود و محدودیت‌های اعمال شده بر پروژه.

مدیر پروژه باید یک رهیافت نظام‌مند، برای انتخاب میزان دقت مناسب برای یک پروژه خاص، در نظر بگیرد. برای انجام این کار، معیارهای تطبیق پروژه تعریف شده‌اند و ارزش یک انتخاب‌گر مجموعه وظایف تخمین زده شده است.

۷-۳-۲ تعریف معیار تطابق

معیارهای تطبیق^۱ برای تعیین میزان دقت توصیه شده‌ای به کار می‌روند که با آن فرایند نرم‌افزاری بر روی پروژه باید به اجرا درآید. یازده معیار تطبیق برای پروژه‌های نرم‌افزاری [PRE 99]^۲ تعریف شده است:

- اندازه پروژه
- تعداد کاربران بالقوه
- حساسیت کار
- دیرپایی کاربرد
- پایایی نیازمندی‌ها
- سهولت ارتباط مشتری/برنامهنویس
- تکامل فناوری قابل کاربرد
- قیود و محدودیتهای عملکرد
- ویژگی‌های جاسازی/غیر جاسازی شده
- کارکنان پروژه
- فاکتورهای مهندسی مجدد



مدل فرایند تطبیقی

به هر یک از معیارهای تطبیق یک شماره اختصاص داده شده است که بین ۱ تا ۵ در تغییر است، در این حالت ۱، نماینده پروژه‌ای است که در آن یک مجموعه فرعی کوچک از وظایف فرایند مورد نیاز است و نیازمندی‌های روش‌شناختی و مستندسازی کلی در حداقل هستند، و ۵، نماینده پروژه‌ای است که در آن یک مجموعه کامل از وظایف فرایند باید به اجرا درآید و نیازمندی‌های روش‌شناختی و مستندسازی قابل ملاحظه هستند.

1. Adaptation criteria

2. Pressman, R. S.

معیار انطباق	درجه	وزن	ضرب امتیاز ورودی					حاصل
			پروژه‌های توسعه مقعمر	پروژه‌های کاربرد جدید	پروژه‌های پیشرفت	پروژه‌های نگهداری	پروژه‌های مهندسی مجدد	
سایز پروژه	—	۱.۲۰	۰	۱	۱	۱	۱	—
تعداد کاربران	—	۱.۱۰	۰	۱	۱	۱	۱	—
میزان بحران تجاری	—	۱.۱۰	۰	۱	۱	۱	۱	—
طول عمر	—	۰.۹۰	۰	۱	۱	۱	۰	—
پایداری نیازمندیها	—	۱.۲۰	۰	۱	۱	۱	۱	—
سهولت ارتباطات	—	۰.۹۰	۱	۱	۱	۱	۱	—
بلوغ فناوری	—	۰.۹۰	۱	۱	۰	۰	۱	—
قیود عملکردی	—	۰.۸۰	۰	۱	۱	۱	۰	—
جاسازی شده / نشده	—	۱.۲۰	۱	۱	۱	۱	۱	—
کارگزینی پروژه	—	۱.۰۰	۱	۱	۱	۱	۱	—
قابلیت عملیات درونی	—	۱.۱۰	۰	۱	۱	۱	۱	—
عوامل مهندسی مجدد	—	۱.۲۰	۰	۰	۰	۰	۱	—
انتخاب‌گر مجموعه وظایف (TSS)								

جدول ۱-۷ محاسبات انتخاب‌گر مجموعه وظایف (TSS)

۳-۳-۷ محاسبه یک معیار انتخاب مجموعه وظایف

برای انتخاب مجموعه وظایف مناسب برای یک پروژه، مراحل زیر باید انجام گیرند:

۱. هر یک از معیارهای تطبیق بخش ۳-۳-۷ را بررسی کنید و بر مبنای ویژگی‌های

پروژه شماره مناسب (۱ تا ۵) را اختصاص دهید. شماره‌ها باید وارد جدول ۱-۷ گردند.

۲. عامل‌های اصلی مختص هر معیار را بررسی کنید. ارزش هر عامل اصلی بین ۰/۸

تا ۱/۲ در تغییر است و نشانه‌ای از اهمیت نسبی یک معیار تطبیق خاص نسبت به نوع

نرم‌افزاری که در محیط محلی توسعه یافته است، به دست می‌دهد. اگر انجام اصلاحات و

تعدیلاتی برای بهتر شدن موقعیت‌های محلی لازم باشد، می‌توان آنها را انجام داد.

۳. شماره‌هایی را که وارد جدول ۱-۷ گردیده‌اند در عامل وزن^۱ و ضرب نقطه

ورودی^۲ پروژه انتخابی ضرب نمایید. ضرب نقطه ورودی دارای ارزش صفر یا یک است و

رابطه معیار تطبیق به نوع پروژه را نشان می‌دهد. نتیجه حاصل ضرب:

ضرب نقطه ورودی × عامل وزن × شماره (گردد)

در ستون حاصل ضرب جدول ۱-۷ برای هر معیار تطبیق جداگانه آورده شده است.



چگونه می‌توانیم

مجموعه وظایف

مناسبی برای پروژه

خود انتخاب کنیم؟

1. weighting factor

2. entry point multiplier

۴. میانگین همه مدخل‌های ستون حاصل ضرب را محاسبه کنید و نتیجه را در قسمت انتخاب‌گر مجموعه وظایف (TSS)^۱ قرار دهید. این ارزش (مقدار) به شما کمک می‌کند تا مجموعه وظایفی را انتخاب کنید که برای پروژه از بقیه مناسب‌تر است.

ضریب امتیاز ورودی					درجه	وزن	معیار انطباق
پروژه‌های مهندسی	نگهداری	پیشرفت	کاربرد	پروژه‌های توسعه			
—	—	—	۱	—	۱.۲	۲	سایز پروژه
—	—	—	۱	—	۱.۱	۳	تعداد کاربران
—	—	—	۱	—	۱.۱	۴	میزان بحران تجاری
—	—	—	۱	—	۰.۹	۳	طول عمر
—	—	—	۱	—	۱.۲	۲	پایداری نیازمندیها
—	—	—	۱	—	۰.۹	۲	سهولت ارتباطات
—	—	—	۱	—	۰.۹	۲	بلوغ فن‌آوری
—	—	—	۱	—	۰.۸	۳	قیود عملکردی
—	—	—	۱	—	۱.۲	۳	جاسازی شده / نشده
—	—	—	۱	—	۱.۰	۲	کارگزینی پروژه
—	—	—	۱	—	۱.۱	۴	قابلیت عملیات درونی
—	—	—	۰	—	۱.۲	۰	عوامل مهندسی مجدد

انتخاب‌گر مجموعه وظایف (TSS)

جدول ۷-۲ مثالی از محاسبات انتخاب‌گر مجموعه وظایف (TSS)

۷-۲-۴ تفسیر مقدار انتخاب‌گر مجموعه وظایف (TSS) و انتخاب مجموعه وظایف

زمانی که انتخاب‌گر مجموعه وظایف محاسبه شد، با استفاده از رهنمون‌های ذیل می‌توان مجموعه کار

مناسب پروژه را انتخاب کرد:

ارزش انتخاب‌گر مجموعه کار

$$TSS < 1/2$$

$$1/10 < TSS < 3/10$$

$$TSS > 2/4$$

میزان دقت

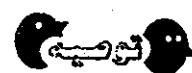
اتفاقی

ساختاری

ثابت

هم‌پوشی ارزش‌های TSS از یک مجموعه وظایف خاص به دیگری هدفمند است، و نشان‌دهنده این

است که به‌هنگام انتخاب مجموعه وظایف نمی‌توان مرزهای مشخصی را تعریف کرد. در تجزیه و تحلیل



اگر مقدار انتخاب شده

مجموعه وظایف در

ناحیه مشترک باشد

(برای مثال ۱ تا ۱.۲ -

مترجم) معمولاً مناسب

است که درجه رسمی

کمتر را لحاظ کنیم، مگر

آنکه ریسک پروژه بالا

نهایی، ارزش انتخاب‌گر مجموعه وظایف، تجربه گذشته و عقل سلیم همه باید در انتخاب مجموعه کار پروژه لحاظ شوند.

جدول ۲-۷ نشان می‌دهد که چگونه TSS را می‌توان برای یک پروژه فرضی محاسبه کرد. مدیر پروژه شماره‌ها را از ستون شماره انتخاب می‌کند. نوع پروژه توسعه کاربرد جدید^۱ می‌باشد. بنابراین ضریب نقطه ورودی از ستون پروژه‌های کاربرد جدید انتخاب می‌گردد. مدخل ستون حاصل ضرب به طریق زیر محاسبه می‌شود:

ضریب نقطه ورودی توسعه جدید \times وزن \times شماره

ارزش TSS (که به عنوان میانگین همه مدخل‌های ستون حاصل ضرب محاسبه می‌شود) ۲/۸ است. با استفاده از معیارهایی که در بالا بحث گردید، این امکان برای مدیر وجود دارد که مجموعه وظایف ساخت‌یافته یا ثابت را به کار ببرد. تصمیم نهایی زمانی اتخاذ می‌گردد که همه فاکتورهای پروژه مورد ملاحظه قرار گرفته باشند.

۴-۷. انتخاب وظایف مهندسی نرم افزار

به منظور در دست داشتن یک برنامه زمان‌بندی برای پروژه، مجموعه وظایف باید در مدت زمانی پروژه توزیع گردد. همان‌طور که در بخش ۳-۷ ذکر کردیم، مجموعه وظایف نسبت به نوع پروژه و میزان دقت تغییر خواهد کرد. هر یک از انواع پروژه‌هایی که در بخش ۳-۷ توصیف شده است با به کار بردن یک مدل فرآیندی که خطی، متوالی و تکراری (به عنوان مثال مدل‌های افزایشی یا نمونه‌سازی) و یا تکاملی است (مانند مدل حلزونی) قابل دسترسی باشند، در بعضی موارد یک پروژه به آرامی به پروژه بعدی ملحق می‌شود. به عنوان مثال، پروژه‌های توسعه مفهوم موفق اغلب وارد پروژه‌های توسعه کاربرد جدید می‌شوند، بعضی وقت‌ها با پایان یافتن یک پروژه توسعه کاربرد جدید، یک پروژه افزایش کاربرد شروع می‌گردد. این پیشرفت طبیعی و قابل پیشگویی است و بدون توجه به مدل فرآیندی که توسط یک سازمان در پیش گرفته شده است اتفاق می‌افتد. بنابراین وظایف مهندسی نرم‌افزار اصلی که در بخش‌هایی که به دنبال می‌آیند توصیف شده‌اند، در مورد تمام مدل‌های فرآیندی قابل اجرا هستند. به عنوان یک مثال، ما وظایف مهندسی نرم‌افزار برای یک پروژه توسعه مفهوم را مورد ملاحظه قرار می‌دهیم:

تعیین دامنه مفهوم - توانایی سازمان برای انجام کاری که در دامنه پروژه نهفته است را به وجود می‌آورد

طرح ریزی اولیه مفهوم - برقرارساختن توانایی سازمان در به عهده گرفتن کارهای مربوط به حوزه

پروژه

ارزیابی ریسک فنی - خطری را که همراه فناوری مورد کاربرد حوزه پروژه است را ارزیابی می‌کند.



یگ مدل فرآیند

نظیفی APM

مشمول بر مجموع

متموعی از وظایف است

که برای استفاده شما

در دسترس می‌باشد.

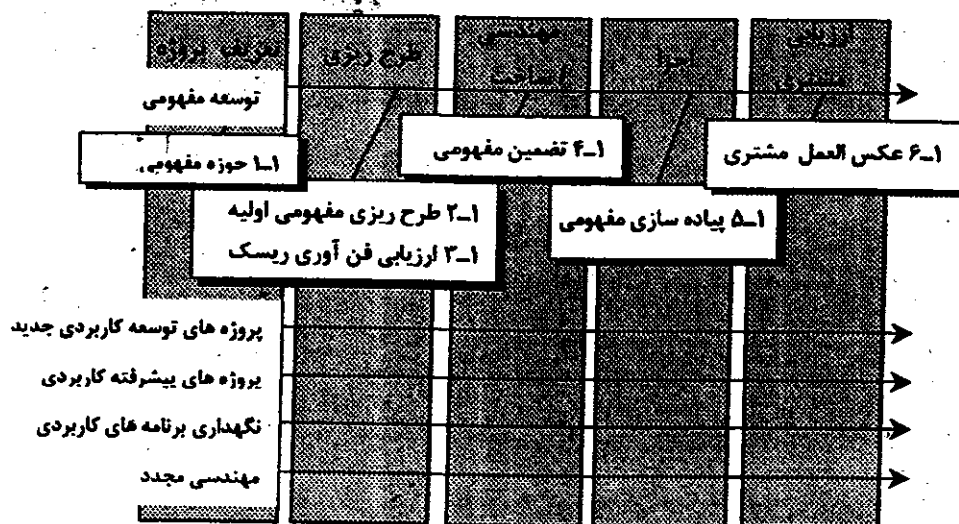
آزمون مفهوم - کارایی یک فناوری نو را حوزه نرم‌افزاری نشان می‌دهد.

پیاده‌سازی مفهوم - نمود مفهوم را به گونه‌ای که توسط یک مشتری قابل بررسی باشد پیاده می‌کند؛ و برای اهداف بازاریابی به کار می‌رود، زمانی که در نظر باشد مفهومی به مشتریان یا مدیریت دیگری فروخته شود.

واکنش مشتری به مفهوم - واکنش نسبت به مفهوم یک فناوری نو را جستجو می‌کند و کاربردهای مشتری خاصی را مورد هدف قرار می‌دهد.

نگاهی سریع به این کارها موجب کمی شگفتی خواهد شد. در واقع، حرکت مهندسی نرم‌افزار به‌سوی پروژه‌های توسعه مفهوم (و نیز تمام انواع دیگر پروژه‌ها) تا اندازه کمی از عقل سلیم بیشتر است.

تیم نرم‌افزار می‌باید بداند که چه کارهایی باید انجام گیرد (تعیین حوزه): تیم (یا مدیر) باید تعیین کند که آیا کسی برای انجام آن وجود دارد (برنامه‌ریزی): خطرات ممکن را مورد ملاحظه قرار دهد (تخمین خطر): فناوری را تا اندازه‌ای مورد تعمق قرار دهد (آزمون مفهوم) و آن را به گونه‌ای شاخص پیاده نماید به‌طوری‌که مشتری بتواند آن را ارزیابی کند (پیاده‌سازی مفهوم و ارزشیابی مشتری) و در پایان، اگر مفهوم کارآمد است، یک‌گونه تولیدی (تبدیلی) باید تولید گردد.



شکل ۷-۱ وظایف توسعه مفهومی در یک مدل خطی

ذکر این نکته حائز اهمیت است که فعالیت‌های شبکه وظایف توسعه مفهوم ماهیتاً تکراری هستند. یعنی یک پروژه توسعه مفهوم واقعی از راه تعدادی افزایش برنامه‌ریزی شده به این فعالیت‌ها نزدیک می‌شود که هر یک مختص تولید یک محصول قابل تحویل، که توسط مشتری قابل ارزشیابی است، هستند.

اگر یک مدل فرآیند خطی انتخاب شده باشد، همان‌طور که نمودار ۷-۱ نشان می‌دهد هر یک از این افزایش‌ها در یک توالی تکراری تعریف شده‌اند. در خلال هر مرحله، فعالیت‌های پوششی (که در فصل ۲ توصیف شدند) اجرا می‌شوند؛ کیفیت کنترل می‌شود، و در پایان هر مرحله یک محصول قابل تحویل تولید می‌شود. با هر تکرار، محصول قابل تحویل باید به محصول تعریف شده پایانی در مرحله توسعه مفهوم نزدیک‌تر شود. اگر یک مدل تکاملی انتخاب شده باشد، طرح‌بندی کارهای ۱/۱ تا ۱/۶ همان‌طور که در نمودار ۷-۲ نشان داده شده است ظاهر خواهد گردید. وظائف بزرگ مهندسی نرم‌افزار برای انواع دیگر پروژه‌ها نیز به شیوه مشابهی قابل تعریف و اجراست.

۷-۵ پالایش وظائف اصلی

کارهای بزرگی که در بخش ۷-۴ توصیف گردیدند ممکن است برای تعریف یک برنامه زمان‌بندی ماکروسکوپی برای یک پروژه به کار روند. با این وجود، برنامه زمان‌بندی ماکروسکوپی برای ایجاد یک برنامه جزئی برای پروژه باید پالایش گردد. پالایش با در نظر گرفتن کارهای بزرگ و تجزیه آنها به کارهای فرعی (با محصولات کاری و هدف‌های اصلی مربوط) شروع می‌شود.

به عنوان نمونه‌ای از تجزیه کردن وظیفه، تعیین حوزه برای یک پروژه توسعه مفهوم^۱ را، که در بخش ۷-۴ بحث شد، مورد ملاحظه قرار دهید. پالایش وظیفه با به کار بردن یک طرح خلاصه شده قابل انجام است، اما در این کتاب، یک رهیافت زبان طراحی فرآیندی برای نشان دادن جریان فعالیت تعیین حوزه مفهوم به کار برده می‌شود:



مدل فرآیند قابل

تطبیق (APM)

مشتمل بر یک توصیف

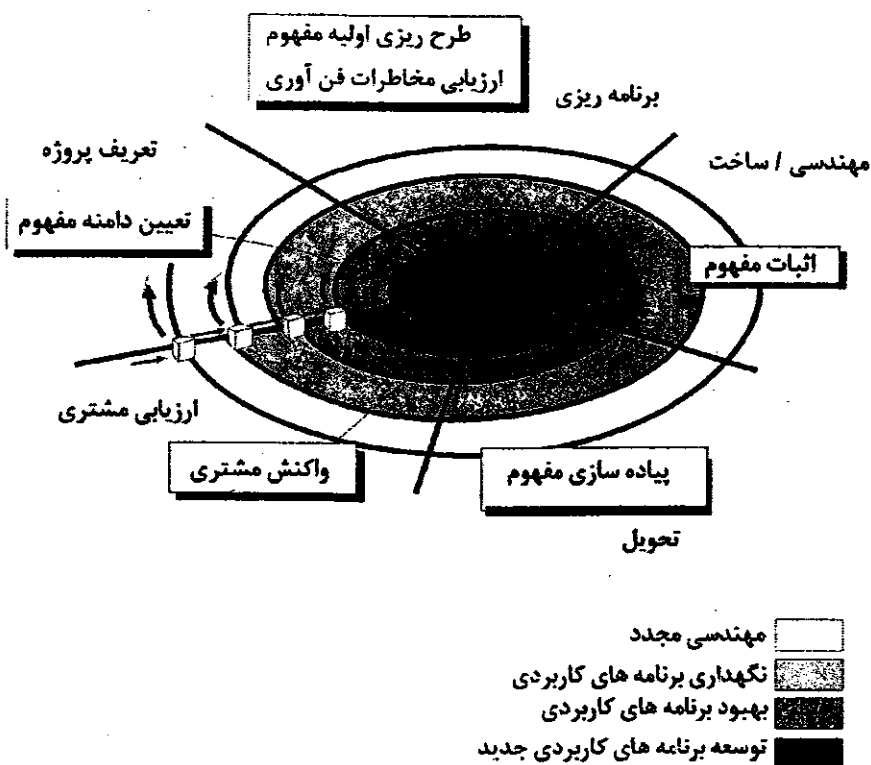
زبان طراحی فرآیند

کاملی است که برای

تمام که برای تمام

وظایف مهندسی

نرم‌افزار کاربرد دارد.



شکل ۷-۲ وظایف توسعه مفهوم با استفاده از مدل تکاملی (افزایشی)

تعریف وظیفه : وظیفه I-۱ تعیین حوزه مفهوم

I-۱-۱ نیازها، منافع و مشتریان بالقوه را شناسایی کنید.

I-۱-۲ رویدادهای برونی کنترل و درونی را که کاربردها را هدایت می کنند، تعریف کنید.

وظیفه I-۱-۲ را شروع کنید.

I-۱-۲-۱ FTR توصیف نوشتاری نیازها را بررسی کنید.^۱

I-۱-۲-۲ لیستی از خروجیها/ورودیهای قابل مشاهده مشتری بدست دهید.

برحسب مورد: دستگاهها

دستگاهها = کیفیت و آمادگی کارکردی

با مشتری ملاقات نمایید تا نیازمندیهای مفهومی اصلی مشخص گردند.

با مصرف کنندگان نهایی مصاحبه کنید؛

رهیافت کنونی به مسئله و روند جاری را مورد مشاهده قرار دهید؛

تقاضاها و شکایات گذشته را بررسی کنید؛

دستگاهها = تجزیه و تحلیل ساخت یافته.

لیستی از اشیاء داده‌های اصلی تهیه کنید.

۱. FTR. به معنای بازبینی رسمی فنی (فصل ۸) می باشد.

رابطه بین اشیاء را تعریف کنید.

ویژگی‌های اشیاء را تعریف کنید.

دستگاه‌ها: دید عینی

لیستی از کلاسهای مسایل تهیه کنید.

سلسله مراتب کلاس و ارتباطات کلاس را توسعه دهید.

ویژگی‌های کلاسها را توصیف کنید.

پایان مورد

I-۱-۲-FTR۳ خروجیها/ورودیها را با مشتری بررسی کنید و هر جا لازم است آنها را اصلاح کنید؛

وظیفه نهایی - وظیفه I-۱-۲

I-۱-۳-۱ کارکردی/ رفتاری را که هر تابع اصلی اجرا می‌کند تعریف کنید؛

وظیفه I-۱-۳ را آغاز کنید

I-۱-۳-۱-۱ هدفهای خروجی و ورودی را که از وظیفه I-۱-۲ حاصل شدند بررسی کنید؛

I-۱-۳-۲-۱ مدلی را از کارکردها/ رفتارها بدست آورید؛

مورد: دستگاه‌ها

دستگاه‌ها: کیفیت و آمادگی کارکردی

با مشتری ملاقات نمایید و نیازهای مفهومی اصلی را بررسی کنید؛

با مصرف‌کنندگان نهایی مصاحبه کنید؛

رهیافت کنونی به مسئله و روند جاری را مورد مشاهده قرار دهید؛

یک طرح سلسله مراتبی از کارکردها/ رفتارها بدست دهید؛

دستگاه‌ها: تجزیه و تحلیل ساختاری

نموداری از سطح بافتی جریان داده‌ها تهیه کنید؛

نمودار جریان داده‌ها را به‌منظور بدست دادن اطلاعات بیشتر اصلاح کنید؛

با کمترین اصلاح ممکن توضیحات فرآیندی برای کارکردها بنویسید؛

دستگاه‌ها: دید عینی

عملیات/ روش‌های مربوط به هر طبقه را تعریف کنید؛

پایان مورد

I-۱-۳-۳-۱ کارکردها/ رفتارها را با مشتری مرور کنید و در موارد لازم آنها را اصلاح کنید؛

وظیفه پایانی - وظیفه I-۱-۳

I-۱-۴ آن عناصر فناوری را که در نرم‌افزار اجرا می‌شوند جدا کنید؛

I-۵-۱ در دسترس بودن نرم‌افزار موجود را مورد بررسی قرار دهید؛

I-۶-۱ امکان‌پذیری فنی را تعریف کنید؛

I-۷-۱ برآوردی سریع از اندازه [کار] بدست دهید؛

I-۸-۱ تعریفی از دامنه کار بدست دهید؛

تعریف وظیفه پایانی: کار I-۱

وظائف و وظائف فرعی‌ای که در پالایش زبان طراحی فرآیند به آنها اشاره شد، پایه یک برنامه زمان‌بندی جزئی، برای فعالیت تعیین حوزه مفهوم، را تشکیل می‌دهند.

۶-۷ تعریف یک شبکه وظائف

وظائف و وظائف فرعی بر پایه توالی‌شان، به هم وابستگی‌های متقابل دارند. به علاوه هرگاه بیش از یک نفر در یک پروژه مهندسی نرم‌افزاری درگیر باشند، احتمال اجرای موازی فعالیت‌ها و وظائف توسعه‌ای وجود دارد. هرگاه چنین چیزی رخ دهد، کارهای هم‌رود باید با هم هماهنگ شوند، به‌طوری‌که وقتی کارهای بعدی تولید کاری آنها را نیاز دارد آنها کامل شده باشند.

یک شبکه وظائف^۱، که شبکه فعالیت^۲ نیز گفته می‌شود، نمایشی نموداری از جریان وظیفه برای پروژه می‌باشد. بعضی وقت‌ها از آن به‌عنوان مکانیسمی استفاده می‌کنند که از راه آن توالی وظیفه و وابستگی‌های آن به یک ابزار برنامه‌ریزی خودکار پروژه انتقال می‌یابند. در ساده‌ترین شکل آن (که برای ایجاد یک برنامه ماکروسکوپی به کار می‌رود)، شبکه وظیفه در توصیف کارهای اصلی مهندسی نرم‌افزار به کار می‌رود. تصویر ۳-۷ یک شبکه وظائف کار شماتیک را برای یک پروژه توسعه مفهوم نشان می‌دهد.



"شبکه وظائف"

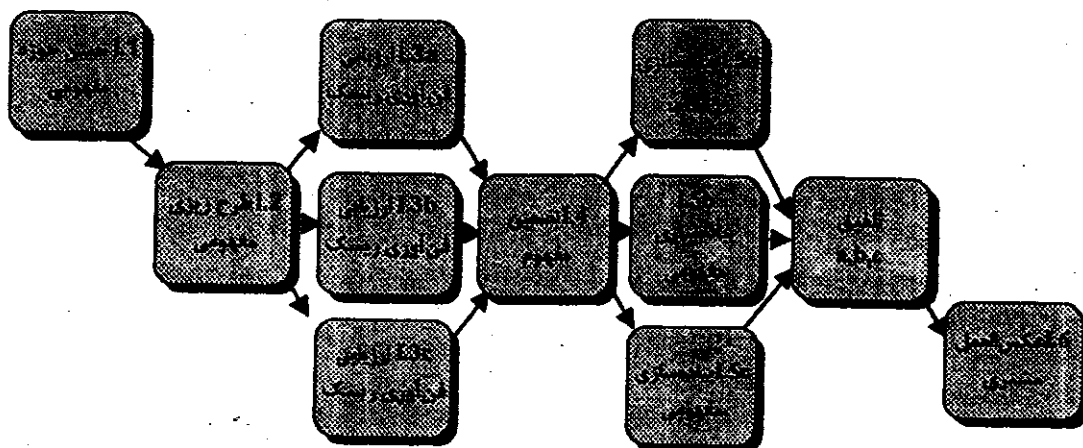
مکانیسمی مفید برای

ترسیم و تشریح

وابستگی‌های بین

وظائف و تعیین مسیر

بحرانی می‌باشد.



شکل ۳-۷

یک شبکه کاری (وظیفه‌ای) برای توسعه مفهوم

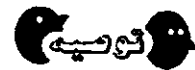
سه وظیفه I.5 موازی
برای سه فانکشن مفهومی
مختلف بکار می‌رود

1.task network

2.activity network

ماهیت هم‌زمانی فعالیت‌های مهندسی نرم‌افزاری منجر به نیازمندیهای برنامه‌ریزی مهمی می‌شود. از آن‌جا که کارهای موازی هم‌زمان اتفاق می‌افتد، برنامه‌ریز باید وابستگی‌های درون - وظیفه را مشخص نماید تا پیشرفت مداوم در جهت تکمیل کار تضمین گردد. به‌علاوه، مدیر پروژه باید از وظائفی که در موقعیت بحرانی قرار می‌گیرند آگاه باشد. یعنی، آن کارهایی که در برنامه زمان‌بندی باید کامل گردند اگر کل پروژه قرار است در یک برنامه زمان‌بندی شده تکمیل گردد. این موضوعات با جزئیات بیشتری بعداً در این فصل بحث خواهند شد.

لازم به‌ذکر است که شبکه وظائفی که در تصویر ۷-۲ نشان داده شده است ماکروسکوپی است. در یک شبکه وظائف جزئی (که مقدمه‌ای برای یک برنامه زمان‌بندی جزئی است) هر فعالیتی که در تصویر ۷-۲ نشان داده شده است باید بسط داده شود. به‌عنوان مثال وظیفه I-۱ تا آن‌جا بسط داده خواهد شد که تمام وظائفی را که به‌طور جزئی در پالایش وظیفه I-۱ در بخش ۷-۵ دیده شد، نشان دهد.



برای تمام پروژه‌ها، مگر پروژه‌های بسیار ساده زمان‌بندی می‌تواند به کمک ابزار زمان‌بندی صورت پذیرد.

۷-۷ زمان‌بندی

زمان‌بندی یک پروژه نرم‌افزاری تفاوت زیادی با زمان‌بندی هر تلاش مهندسی چند جنبه‌ای دیگری ندارد. بنابراین با اندکی تعدیل ابزارهای زمان‌بندی عمومی پروژه‌ها و نیز فنون آن در مورد پروژه‌های نرم‌افزاری نیز قابل اجرا هستند.

ارزشیابی برنامه و فنون بررسی (PERT)^۱ و روش مسیر بحرانی (CPM)^۲ دو روشی هستند که برای زمان‌بندی پروژه به‌کار می‌روند [MOD83] که برای توسعه نرم‌افزاری نیز قابل کاربردند. این دو تکنیک به‌وسیله اطلاعات حاصل از فعالیت‌های برنامه‌ریزی پروژه قبلی بدست آمده‌اند:

- برآوردهای نیروی انسانی
- تجزیه کارکرد محصول
- انتخاب مدل فرآیند مناسب و مجموعه وظائف
- تجزیه وظائف

با به‌کار بردن یک شبکه وظائف امکان تعریف وابستگی‌های موجود بین وظائف وجود دارد. وظائف که بعضی وقتها به آنها ساختار تجزیه کار پروژه (WBS)^۳ گفته می‌شود، برای کل پروژه یا کارکردهای فردی تعریف می‌شوند. ارزشیابی برنامه و فنون بررسی و نیز روش مسیر بحرانی^۴ به برنامه‌ریز نرم‌افزاری امکان می‌دهد: (۱) مسیر بحرانی را مشخص نماید - زنجیره وظائفی که مدت زمانی پروژه را تعیین می‌کند:

1. Program Evaluation & Review Technique (PERT)

2. Critical Path Method

3. work breakdown structure

4. critical path

(۲) "محتمل‌ترین" برآورد زمانی برای کارهای فردی را با به‌کار بردن مدل‌های آماری انجام دهد: (۳) زمان‌های مرزی را محاسبه کند که به تعریف یک "پنجره" زمانی برای یک وظیفه خاص می‌انجامد.

محاسبه زمان مرزی در زمان‌بندی پروژه نرم‌افزاری می‌تواند بسیار سودمند باشد. مثلاً، تغییری جزئی در طرح یک تابع می‌تواند توسعه بیشتر توابع دیگر را به تأخیر اندازد.

ریگز [RIG81]^۱ زمان‌های مرزی مهمی را که ممکن است از ارزشیابی برنامه و فنون بررسی و روش مسیر بحرانی تمیز داده شوند توصیف می‌کند: (۱) اولین زمانی که یک وظیفه بعد از تکمیل تمام وظیفه‌های قبلی، در کوتاه‌ترین زمان ممکن شروع می‌شود: (۲) دیرترین زمان برای شروع وظیفه قبل از به تأخیر افتادن زمان حداقل تکمیل پروژه: (۳) اولین زمان پایانی وظیفه - مجموع اولین شروع و مدت زمانی وظیفه (۴) آخرین زمان پایان وظیفه - آخرین زمان شروع وظیفه به‌علاوه مدت زمان انجام وظیفه (۵) تعلیق کلی^۲ - مقدار زمان افزوده یا فرعی‌ای که در زمان‌بندی وظیفه‌ها لحاظ گردیده است، به‌طوری‌که مسیر بحرانی شبکه در برنامه زمان‌بندی نگه داشته می‌شود. محاسبه مرزی منجر به تعیین مسیر بحرانی می‌شود و روشی کمی در اختیار مدیر می‌گذارد که با آن می‌تواند به هنگام تکمیل شدن کارها پیشرفت آنها را ارزشیابی کند.

هم ارزشیابی برنامه و فنون بررسی و هم روش مسیر بحرانی در انواع گسترده‌ای از ابزارهای خودکار به‌کار برده شده‌اند: که برای بسیاری از کامپیوترهای شخصی در دسترس می‌باشند [THE93]^۳. کاربرد این ابزارها آسان است و روش‌های زمان‌بندی که در بالا توصیف شدند را در دسترس هر مدیر پروژه نرم‌افزاری قرار می‌دهد.

۷-۷-۱ نمودارهای خطی زمانی

به هنگام ایجاد یک برنامه زمان‌بندی برای پروژه نرم‌افزاری برنامه‌ریز آن کارش را با مجموعه‌ای از وظائف (ساختار تقسیم کار) شروع می‌کند. اگر ابزارهای خودکار به‌کار روند، تقسیم کار به‌عنوان شبکه وظیفه یا خلاصه وظیفه در نظر گرفته می‌شود. نیروی کاری، مدت زمانی، و زمان شروع وظیفه و ورودی هر وظیفه در نظر گرفته می‌شوند. همچنین، ممکن است وظائف با افراد خاصی اختصاص داده شوند. در نتیجه این کار، یک نمودار زمانی^۴ که نمودار گانت^۱ نیز گفته می‌شود به‌وجود می‌آید. یک نمودار زمانی را می‌توان به کل پروژه توسعه داد. از طرف دیگر، می‌توان نمودارهای جداگانه نیز برای هر یک از توابع پروژه یا برای کارهای فردی که بر روی پروژه انجام می‌شوند به‌وجود آورد.



ابزارهای CASE

برای پروژه / زمان

بندی و طرح ریزی



یک نمودار خطی، شما

را قادر می‌سازد که

کدام کارها در کدام

نقطه زمانی باید

هدایت و اداره شوند.

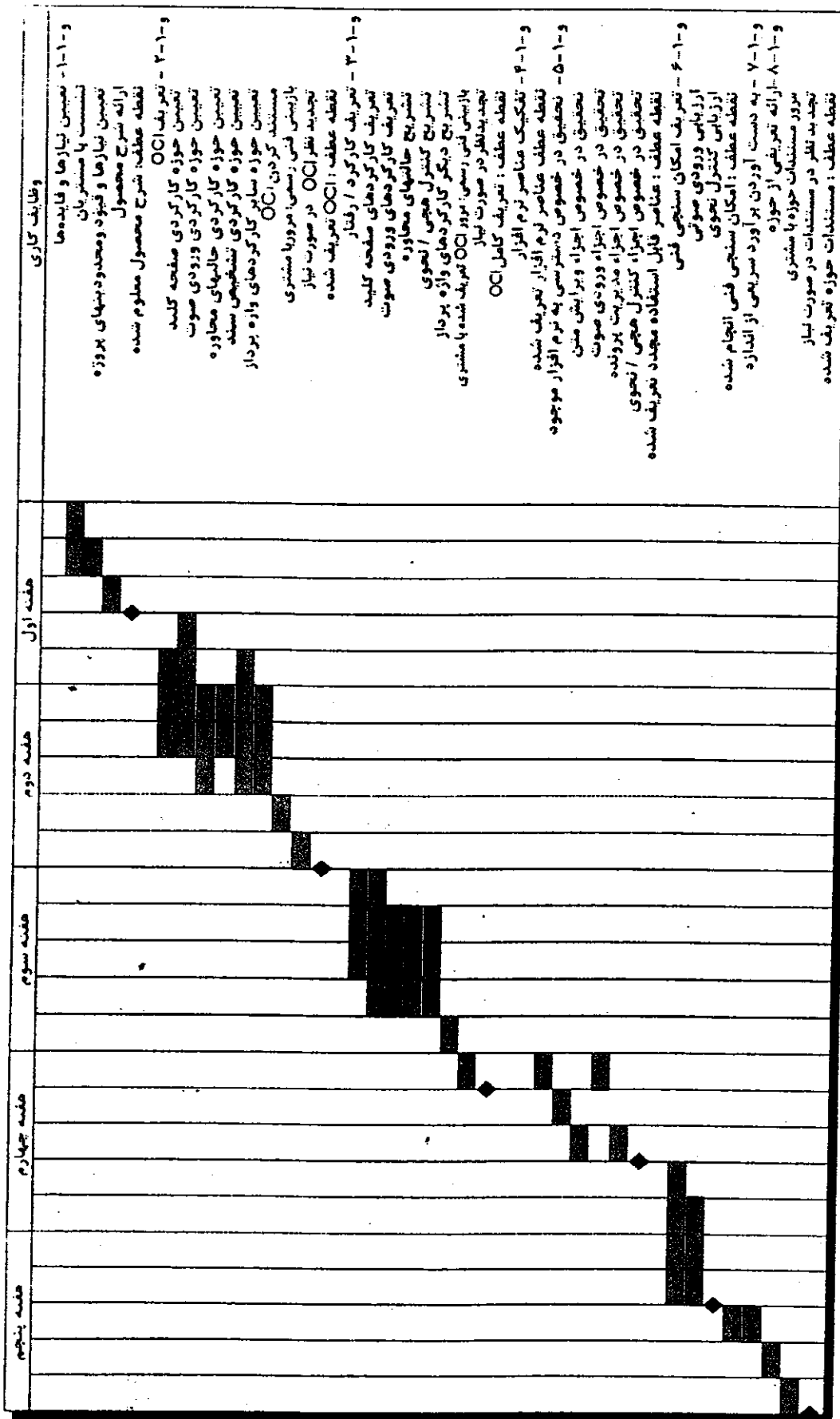
1. Riggs, J.

2. total float

3. The 'L.

4. timeline chart

نمودار ۴-۷ طرح یک نمودار زمانی را نشان می‌دهد. این نمودار قسمتی از برنامه زمانی یک پروژه نرم‌افزاری را نشان می‌دهد، که تکیه بر تعیین حوزه مفهومی وظیفه (بخش ۷-۵) برای یک نرم‌افزار ویژه‌پرداز دارد. تمام کارهای پروژه (برای تعیین حوزه مفهومی) در ستون سمت چپ لیست گردیده‌اند. باریک‌های افقی مدت زمانی هر وظیفه را نشان می‌دهند. وقتی چندین باریکه در یک زمان اتفاق می‌افتند، نشان‌دهنده هم‌زمانی کارهاست. لوزی‌ها نشان‌دهنده نقاط عطف اصلی‌اند.



و : وظیفه
OCI : Output/Control/Input

شکل ۷-۴ مثالی از نمودار خطی زمانی (گانت چارت))

یکبار اطلاعات لازم برای به وجود آوردن یک نمودار زمانی وارد گردیده است، و اکثریت ابزارهای زمانبندی پروژه های نرم افزاری جدولهای پروژه های تولید کرده اند - که لیستی جدول وار از تمام وظائف پروژه، شروع برنامه ریزی شده و واقعی و زمانهای پایان کار، و انواعی از اطلاعات مربوطه میباشد. جدولهای پروژه که به همراه جدولهای زمانی به کار می روند به مدیر پروژه امکان می دهد که پیشرفت کار را پی گیری کند.

ملاحظات	تخصیص نیروی کار	تفصیلی یافته	تکمیل واقعی	تکمیل طرح ریزی شده	شروع واقعی	شروع طرح ریزی شده	وظائف کاری
تعیین حوزه نیازمند اختصاصی زمان و نیروی کار بیشتری است.	2 p-d	BLS JPP	2.0	2.0	1.0	1.0	1-1-1 تعیین نیازها و منابع ملاقات با مشتریان تعیین نیازها و لیست محدودیتهای پروژه تعیین ویژگیهای محصول معیار: ویژگیهای تعریف شده محصول 3-1-1 تعریف و . ک . خ . ه (OCI) مورد نظر
	1 p-d	BLS/JPP	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ساخته کلید
	1 p-d		2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی صوت
	1.5 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه حالات مختلف محاورهای
	2 p-d	JPP	2.0	2.0	2.0	2.0	تعیین حوزه شناسایی مستندات
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه دیگر فانکشنهای واژه پرداز
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی / خروجی (OCI) مستند سازی ورودی / کنترل / خروجی (OCI)
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی / خروجی (OCI) توسط مشتری
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی / خروجی (OCI) توسط مشتری
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی / خروجی (OCI) توسط مشتری
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی / خروجی (OCI) توسط مشتری
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی / خروجی (OCI) توسط مشتری
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی / خروجی (OCI) توسط مشتری
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی / خروجی (OCI) توسط مشتری
	3 p-d	BLS	2.0	2.0	2.0	2.0	تعیین حوزه فانکشنهای ورودی / خروجی (OCI) توسط مشتری

هفته - روز

ورودی / کنترل / خروجی (OCI) / Input / Control / Output
[Formal Technical Review] FTR رسمی فنی بازبینی

شکل ۵-۷ مثالی از جدول پروژه

۲-۷-۷ ردگیری برنامه زمان‌بندی

برنامه زمان‌بندی شده پروژه یک نقشه زمینی برای مدیر پروژه فراهم می‌آورد. اگر آن را به‌طور مناسب تهیه کرده باشند، برنامه زمان‌بندی پروژه وظائف و نقاط عطف اصلی‌ای را که باید در جریان پیشرفت پروژه پی‌گیری و کنترل شوند تعریف می‌کند. پی‌گیری می‌تواند به‌صورت‌های گوناگونی انجام گیرد:

- برگزاری گردهم‌آیی‌های دوره‌ای در مورد وضعیت پروژه که در آن اعضای هر تیم پیشرفت‌ها و مشکلات را گزارش می‌دهد.
- ارزشیابی نتایج تمام بررسی‌هایی که در طول روند مهندسی نرم‌افزاری انجام شده‌اند.
- تعیین این‌که آیا هدف‌های رسمی (نقاط عطف) پروژه (لوزی‌های نشان داده‌شده در نمودار ۴-۷) در برنامه زمانی محقق شده‌اند یا نه؟
- مقایسه شروع واقعی و شروع برنامه‌ریزی شده برای هر یک از وظائف پروژه که در جدول اصلی لیست شده‌اند (نمودار ۵-۷):
- ملاقات غیر رسمی با کارورزان، به‌منظور دستیابی به ارزیابی شخصی آنها از پیشرفت زمانی و یا مسایلی که در کار دارند، یا خطرات.

• به‌کار بردن ارزش سنجی اکتسابی (بخش ۸-۷) به‌منظور ارزیابی کمی پیشرفت‌ها.

یک مدیر پروژه نرم‌افزاری به کنترل پروژه می‌پردازد تا منابع پروژه را اداره نماید، موقعیت را با مسائل سازگار نماید، و کارکنان پروژه را هدایت نماید. اگر کارها به خوبی در حال انجام باشند (مثلاً اگر پروژه بر اساس برنامه و متناسب با بودجه باشد، بررسی نشان می‌دهند که پیشرفت واقعی انجام شده است و هدف‌های اصلی محقق شده‌اند)، کنترل کمتر خواهد بود. اما هرگاه مشکلاتی پیش آیند، مدیر پروژه باید برای رفع آنها هر چه زودتر به اعمال کنترل بپردازد. بعد از این‌که مشکلی تشخیص داده شد،^۱ منابع بیشتری ممکن است به آن زمینه مشکل‌ساز اختصاص داده شود، ترتیب و استخدام کارکنان تغییر کند یا برنامه زمان‌بندی پروژه اصلاح گردد.

در مواردی که فشار زمانی حادی به‌وجود می‌آید، مدیران با تجربه اقدام به‌کارگیری برنامه زمان‌بندی و تکنیک کنترل خاصی در پروژه می‌کنند که به آن جعبه‌ای کردن زمان^۲ گویند. [ZAH95]^۳ راهبرد جعبه‌ای کردن زمان تشخیص می‌دهد که محصول کامل در موعد از قبل تعیین شده قابل عرضه نخواهد بود. بنابراین یک انگاره نرم‌افزاری افزایشی (فصل ۲) انتخاب شده است، و برنامه‌ای برای هر یک از مراحل عرضه افزایشی تنظیم شده است.

۱. توجه به این امر مهم است که خطا در امر زمان‌بندی نتیجه فهم اشتباه از مسئله است. نقش یک مدیر پروژه آن است که فهمی صحیح از مسئله ایجاد نموده، آن را اصلاح کند.

2.time-boxing

3.Zahniser,R.

نقل قول

نخستین قاعده گزارش وضعیت نرم‌افزار می‌تواند در یک جمله خلاصه شود: شکست زده نشوید. کاپرز جونز

توصیه

بهترین معیار پیشرفت، تکمیل و موفقیت در بازبینی محصول کار نرم‌افزاری می‌باشد

بنابراین وظیفه‌ای که به هر یک از مراحل افزایشی اختصاص داده شده است از نظر زمانی به اصطلاح جعبه‌ای شده است. این بدان معناست که برنامه زمانی برای هر وظیفه تنظیم گردیده است و این کار با برگشت از زمان تحویل محصول به مراحل افزایشی انجام گرفته است. یک "جعبه" در اطراف هر وظیفه گذاشته شده است. هر گاه وظیفه‌ای به مرز زمان جعبه‌ای‌اش می‌رسد (مثبت یا منفی ۱۰٪)، آن وظیفه متوقف می‌شود و وظیفه بعدی آغاز می‌گردد.

اغلب واکنش اولیه به رهیافت جعبه‌ای کردن زمان منفی است: "اگر کار به پایان نرسیده است، پس چگونه می‌توانیم پیش برویم؟" جواب این سؤال در چگونگی تکمیل کار نهفته است. زمانی که زمان جعبه‌ای فرا برسد، احتمالاً ۹۰٪ کار کامل شده است.^۱ ۱۰٪ باقی‌مانده، هر چند مهم باشد، تا افزایش بعدی می‌تواند به تعویق بیفتد، یا اگر لازم شد بعداً کامل گردد. در این صورت به جای "در جا زدن" بر روی یک کار، پروژه به‌سوی تاریخ تحویل پیش می‌رود.

۸-۷ تحلیل مقدار بدست آمده (ارزش‌سنجی اکتسابی)

در بخش ۷-۷-۲ ما تعدادی از رهیافتهای کمی پی‌گیری و ردگیری پروژه را مورد بحث قرار دادیم. هر یک از آن رهیافتهای نشانی از پیشرفت را به مدیر پروژه ارائه می‌دهند، اما تخمین اطلاعات ارائه شده تا اندازه‌ای شخصی است. معقول است که بپرسیم آیا تکنیک کمی‌ای برای ارزیابی پیشرفت، به هنگام پیشرفت تیم نرم‌افزاری در کارهایی که به زمان‌بندی پروژه مربوط است، وجود دارد یا ندارد؟ در واقع، تکنیکی برای اجرای تحلیل کمی پیشرفت وجود دارد؛ که نام آن ارزش‌سنجی اکتسابی است (EVA).^۲

هامفری [Hum95]^۳ ارزش اکتسابی را به نحو زیر تعریف می‌کند:

سیستم ارزش اکتسابی یک میزان ارزشی مشترک برای هر [پروژه نرم‌افزاری] وظیفه فراهم می‌کند، صرف‌نظر از نوع وظیفه‌ای که انجام شده است. کل ساعات، انجام تمام پروژه برآورد شده است، و به هر کار بر اساس درصد محاسبه شده کلی یک ارزش اکتسابی داده شده است.

خیلی ساده گفته شده است، که ارزش اکتسابی، واحدی برای پیشرفت است که ما را قادر می‌سازد "درصد کامل بودن" یک پروژه را با به کار بردن تجزیه و تحلیل کمی به جای احساس غریزی ارزیابی کنیم. در واقع، فلمینگ و کابلن [FLE 98]^۴ معتقدند که ارزش‌سنجی اکتسابی "برداشت‌های دقیق و قابل اطمینانی از اجرای پروژه از شروع ۱۵ درصد اولیه آن به بعد فراهم می‌کنند."



مقادیر به دست آمده.

ممیزی کمی برای

پیشرفت می‌های می

سازند.

۱. می‌توان این امر را به گونه ای دیگر گفت: ۹۰ درصد اول یک سیستم، ۱۰ درصد زمان می‌برد و ۱۰ درصد آخر، ۹۰ درصد زمان خواهد برد. (قانون ۹۰-۱۰ بیشتر متداول است. -م)

2. Earned Value Analysis

3. Humphrey, W.

4. Fleming, Q. W. and J. M.

برای تعیین ارزش اکتسابی مراحل ذیل اجرا شده‌اند:

۱- ارزش بودجه‌ای کار زمان‌بندی شده^۱ (BCWS) برای هر وظیفه‌ای که در برنامه

زمان‌بندی آمده است تعیین شده است. در خلال ارزیابی فعالیت‌ها (فصل ۵)، (نفر- ساعت یا نفر- روز) هر وظیفه مهندسی نرم‌افزاری برنامه‌ریزی شده است. بنابراین ارزش بودجه‌ای کار زمان‌بندی شده^۱ تلاشی است که برای کار^۱ برنامه‌ریزی شده است. برای تعیین پیشرفت در نقطه خاصی از زمان‌بندی پروژه، مقدار ارزش بودجه‌ای کار زمان‌بندی شده برابر است با حاصل جمع مقدارهای ارزش بودجه‌ای کار زمان‌بندی شده^۱ که در آن نقطه زمانی در پروژه کامل شده‌اند.

۲- مقدارهای ارزش بودجه‌ای کار زمان‌بندی شده^۲ مقادیر BCWS برای تمام کارها جمع شده‌اند، تا بودجه را در زمان تکمیل پروژه نشان دهند. بنابراین اگر آن را BAC بنامیم داریم:

$$BAC = \sum (BCWS_k) \text{ for all tasks } k$$

برای همه کارهای K (ارزش بودجه‌ای زمان‌بندی کار K) = بودجه در زمان تکمیل

۳- مرحله بعد، ارزش بودجه‌ای کار اجرا شده (BCWP) محاسبه شده است. مقدار ارزش بودجه‌ای کار اجرا شده برابر است با حاصل جمع مقدارهای ارزش بودجه‌ای کار اجرا شده برای همه کارهایی که واقعاً در یک نقطه زمانی در زمان‌بندی پروژه تکمیل شده‌اند.

ویلکنز [WIL99]^۳ اشاره می‌کند که "تفاوت بین ارزش بودجه‌ای کار زمان‌بندی شده و ارزش بودجه‌ای کار اجرا شده در این است که اولی نماینده بودجه فعالیت‌هایی است که برای تکمیل آنها برنامه‌ریزی شده است، در حالی که دومی نماینده بودجه فعالیت‌هایی است که واقعاً کامل شده‌اند." با داشتن مقدارهای ارزش بودجه‌ای کار زمان‌بندی شده، بودجه "تکمیل کار و ارزش بودجه‌ای کار اجرا شده، نشانگرهای عمده پیشرفت را می‌توان محاسبه کرد:

$$\text{Schedule Performance Index, } SPI = BCWP / BCWS$$

ارزش بودجه‌ای کار زمان‌بندی شده / ارزش بودجه‌ای کار اجرا شده = شاخص عملکرد زمان بندی

$$\text{Schedule Varance, } SV = BCWP - BCWS$$

ارزش بودجه‌ای کار زمان‌بندی شده - ارزش بودجه‌ای کار اجرا شده = واریانس زمان بندی

شاخص عملکرد زمان بندی SPI، نشانگر کارایی است که به وسیله آن پروژه منابع برنامه‌ریزی شده را به کار می‌برد. شاخص عملکرد زمان بندی به میزان ۱/۰ نشان‌دهنده اجرای کارآی برنامه پروژه است. واریانس برنامه به سادگی نشان‌دهنده واریانسی از زمان‌بندی برنامه‌ریزی شده است.

$$\text{Percent scheduled for completion} = BCWP / BAC$$



مقادیر بدست آمده را چگونه محاسبه کنیم تا ممیزی برای پیشرفت قلمداد گردند؟

1. budgeted cost of work scheduled (BCWS)

2. The budgeted cost of work performed (BCWP)

3. Wilkens, T. T.

بودجه به هنگام تکمیل کار / ارزش بودجه‌ای کار اجرا شده = درصد برنامه‌ریزی شده برای تکمیل کار که نشانگر کمی از "درصد کامل شدن" پروژه در نقطه زمانی t بدست می‌دهد.

همچنین محاسبه ارزش واقعی کار اجرا شده^۱ (ACWP) ممکن می‌باشد. مقدار ارزش واقعی کار اجرا شده برابر است با حاصل جمع تلاش واقعی‌ای که برای کار صرف شده است، کاری که در یک نقطه زمانی در برنامه پروژه کامل شده است.

بنابراین موارد زیر را می‌توان محاسبه کرد:

$$\text{Cost Performance Index, CPI} = \text{BCWP} / \text{ACWP}$$

ارزش واقعی کار اجرا شده / ارزش بودجه‌ای کار اجرا شده = شاخص اجرایی هزینه

$$\text{Cost Variance, CV} = \text{BCWP} - \text{ACWP}$$

ارزش واقعی کار اجرا شده / ارزش بودجه‌ای کار اجرا شده = واریانس هزینه

شاخص اجرای هزینه‌ای به مقدار ۱/۰ به خوبی نشان می‌دهد که پروژه در جهت بودجه تعریف شده اش در حرکت است. واریانس هزینه نشانه معتبری از درآمد حاصله از صرفه‌جویی در هزینه (برخلاف هزینه‌های برنامه‌ریزی شده) یا کاستی در مرحله خاصی از پروژه می‌باشد. همانند رانداری در افق، ارزش‌سنجی اکتسابی مشکلات زمان‌بندی را قبل از این‌که آنها به‌نحوی ظاهر شوند نشان می‌دهد. که همین مدیر پروژه نرم‌افزاری را قادر می‌سازد قبل از گسترده شدن بحران در پروژه اقدامی اصلاحی انجام دهد.



رشته گسترده ای از منابع تحلیل مقادیر بدست آمده (فهرست جامع، مقالات، اتصالات و ارجاعات) در آدرس زیر وجود دارد:

www.acq.osd.mil/pml/

۷-۹ ردگیری خطا

در روند پروژه نرم‌افزاری، تیم پروژه محصولات کاری خاص خود را دارد (مثلاً، تعیین نیازها یا الگوی اولیه، مستندات طراحی، کد منبع)، اما تیم همچنین خطاهایی نیز به همراه هر محصول کاری مرتکب می‌شود (و خوشبختانه آنها را اصلاح می‌کند). اگر متریک‌های برآیند آنها در بسیاری از پروژه‌های نرم‌افزاری جمع شده باشند، مدیر پروژه می‌تواند این داده‌ها را به‌عنوان مبنای مقایسه، در برابر داده‌های خطاهای انباشته شده در زمان واقعی به‌کار ببرد - ردیابی خطا می‌تواند به‌عنوان وسیله‌ای برای ارزیابی وضعیت یک پروژه در حال اجرا به‌کار رود.

در فصل ۴، مفهوم کارایی و بازده رفع نقص مورد بحث قرار گرفت. به‌طور خلاصه، تیم نرم‌افزاری به اجرای بررسی‌های فنی رسمی (و سپس آزمودن) برای یافتن و اصلاح خطاهای E، که در محصولات کاری در خلال وظائف مهندسی نرم‌افزاری تولید شده‌اند، دست می‌زند. هر خطایی که آشکار نشود (اما در کارهای بعدی یافت شود) به‌عنوان نقص در نظر گرفته می‌شود، D کارایی و بازده رفع نقص (فصل ۴)



جستجوی خطاها به شما اجازه می‌دهد، فعالیت کنونی خود را با نیروی پیشین مقایسه نموده و یک معیار کمی برای کیفیت کاری که اداره می‌شود، تهیه کنید

1. actual cost of work preformed

$$D = E / (E+D)$$

به‌صورت زیر تعریف شده است:

(بازده رفع نقص)

بازده (کارایی) رفع نقص یک متریک پردازشی است که نشانه معتبری از مؤثر بودن فعالیت‌های تضمین کیفیت بدست می‌دهد، اما DRE و شمارش خطاها و نواقص مربوط به آن را می‌توان برای کمک به مدیر برای تعیین پیشرفت حاصل شده در جریان حرکت پروژه نرم‌افزاری در مسیر کار برنامه ریزی شده‌اش به کار برد.

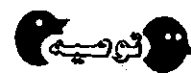
اجازه دهید فرض کنیم یک سازمان نرم‌افزاری اقدام به جمع‌آوری اطلاعات مربوط به خطاها و نقص‌هایش در طول ۲۴ ماه کرده است و برای متریک‌های ذیل میانگین‌های به‌وجود آورده است:

- خطاهای هر صفحه تعیین نیازها، E_{req}
- خطاهای هر سطح طراحی اجزاء، E_{design}
- خطاهای هر سطح برنامه‌نویسی اجزاء، E_{code}
- بازده رفع نقص - تحلیل نیازها
- بازده رفع نقص - طراحی معماری
- بازده رفع نقص - طراحی در سطح اجزاء
- بازده رفع نقص - برنامه‌نویسی

همزمان با پیشرفت‌هایی که پروژه در مراحل مهندسی نرم‌افزار حاصل می‌کند، تیم نرم‌افزاری تعداد خطاهایی را که در جریان بررسی نیازها، طرح‌ها و برنامه‌ها یافت شده‌اند را ثبت و گزارش می‌دهد. مدیر پروژه ارزش‌های جاری E_{req} ، E_{design} و E_{code} را محاسبه می‌کند. و سپس آنها را با میانگین‌های پروژه‌های قبلی مقایسه می‌کند. اگر نتایج جاری بیش از ۲۰٪ از میانگین تجاوز کنند، دلیلی برای تعمق و مطمئناً تحقیق وجود دارد.

به‌عنوان مثال، اگر برای پروژه X ، $E_{req} = 2/1$ باشد در حالی که میانگین سازمانی ۳/۱۶ است، یکی از این دو سناریو ممکن است: (۱) تیم نرم‌افزاری کار قابل ملاحظه‌ای در تعیین نیازها انجام داده است یا (۲) تیم در بررسی‌هایش سهل‌انگار بوده است. اگر احتمال صدق سناریوی دوم می‌رود، مدیر پروژه به‌منظور رفع نقص‌هایی که احتمالاً در جریان طراحی تکثیر شده‌اند، باید قدم‌های سریعی برای به وجود آوردن طرح زمانی^۱ دیگری در برنامه زمان‌بندی بردارد.

متریک‌های ردیابی خطایی که در بالا ذکر شدند همچنین می‌توانند برای بررسی هدف‌ها و / یا در آزمون منابع به کار روند. به‌عنوان مثال اگر سیستمی از ۱۲۰ جزء ترکیب یافته است، اما برای ۳۲ مؤلفه آن



اطلاعات کمی بیشتر

در مورد رهیافت شما

برای ردگیری و کنترل

پروژه، شما را قادر می

سازد که مشکلات

بالقوه را پیش بینی

کرده، راهکار مناسب

بیانید. مقدار

بدست آمده و متریک

های ردگیری را مورد

استفاده قرار دهید.

۱. در واقع، مدت زمانی طولانی باید برای کار مجدد بر روی معایب و نواقص سپری شود، البته این امر در صورت طراحی نامناسب اتفاق خواهد افتاد.

مقدارهای Edesign دارای واریانس عمده‌ای نسبت به میانگین باشند، پس مدیر پروژه ممکن است منابع بررسی که را به آن ۲۲ جزء اختصاص دهد و به اجزاء دیگر اجازه دهد که با یک بررسی کد وارد جریان آزمون شوند. اگر چه در یک آزمون ایده‌آل همه اجزاء باید از جریان بررسی کد بگذرند، اما یک رهیافت گزینشی (یعنی بررسی آن حوزه‌هایی که بر مبنای مقدار Edesign دارای کیفیت مشکوکی هستند) ممکن است وسیله مؤثری برای جبران کردن زمان از دست رفته و/یا صرفه‌جویی هزینه‌ها در پروژه‌ای که بودجه‌اش مصرف شده است باشد.

۷-۱۰ طرح پروژه

هر مرحله روند مهندسی نرم‌افزاری باید دارای نتیجه‌ای قابل بررسی باشد که می‌تواند مبنای مراحل که بعد از آن می‌آیند قرار گیرد. برنامه پروژه نرم‌افزاری^۱ در پایان کارهای برنامه‌ریزی به‌وجود می‌آید؛ که هزینه پایه و اطلاعات زمان‌بندی بدست می‌دهد که در تمام مراحل فرآیند نرم‌افزاری به‌کار خواهند رفت. برنامه پروژه نرم‌افزاری سندی نسبتاً کوتاه است که برای مخاطبان گوناگونی تهیه شده است. این برنامه باید (۱) حوزه و منابع را برای مدیریت نرم‌افزار، کارکنان فنی و مشتری بیان نماید؛ (۲) خطرهای تعریف و فنون مقابله با آنها را بیان کند؛ (۳) هزینه و برنامه را برای بررسی مدیریت تعریف کند؛ (۴) یک رهیافت کلی برای توسعه نرم‌افزاری برای همه افرادی که به پروژه مربوط هستند بدست دهد، و (۵) به‌طور خلاصه بیان کند که چگونه کیفیت تضمین می‌شود و چگونه تغییرات را باید مدیریت کرد. لازم به‌ذکر است که طرح پروژه نرم‌افزاری یک سند ثابت نیست. یعنی، تیم پروژه مرتباً طرح را مورد تجدیدنظر قرار می‌دهند - خطرات، برآوردها، برنامه‌ها و اطلاعات مربوط را هم‌زمان با پیشرفت پروژه و یادگیری‌های بیشتر به‌نگام می‌کنند.



طرح پروژه نرم‌افزاری

۷-۱۱ خلاصه

زمان‌بندی اوج یک فعالیت برنامه‌ریزی است که بخشی اساسی در مدیریت پروژه نرم‌افزاری است. در ترکیب با روش‌ها تخمینی و خطرسنجی، زمان‌بندی تبدیل به یک نقشه زمینی برای مدیر پروژه می‌شود. زمان‌بندی با تجزیه فرآیند شروع می‌گردد. از ویژگی‌های پروژه برای انتخاب یک مجموعه وظایف برای فعالیتی که باید انجام گیرد استفاده می‌شود. شبکه وظایف وابستگی هر وظیفه مهندسی، را به دیگر وظایف و مدت زمانی آن را در پروژه توصیف می‌کند. همچنین از شبکه وظایف برای محاسبه مسیر بحرانی، نمودار زمانی و انواعی از اطلاعات پروژه استفاده می‌گردد. مدیر پروژه با به‌کار بردن برنامه به‌عنوان یک راهنما، می‌تواند به پی‌گیری و کنترل هر یک از مراحل فرآیند نرم‌افزاری بپردازد.

مسایل و نکاتی برای تفکر و تعمق بیشتر

۱-۷ تعیین موعدهای "نامعقول" حقیقی از زندگی در تجارت نرم‌افزار است. اگر با یکی از این

تعیین موعدها روبه‌رو شدید چه واکنشی باید نشان دهید؟

۲-۷ چه تفاوتی بین یک برنامه زمان‌بندی ماکروسکوپی و یک برنامه زمان‌بندی جزئی موجود است؟ آیا ادله کردن یک پروژه در صورتی که تنها دارای یک برنامه زمان‌بندی ماکروسکوپی باشد ممکن است؟ چرا؟

۳-۷ آیا موردی وجود دارد که در آن هدفهای اساسی یک پروژه نرم‌افزاری نیازی به بررسی نداشته باشد؟

۴-۷ در بخش ۱-۲-۷ ما مثالی از "ارتباط صعودی" ارائه کردیم که می‌تواند در صورت کار کردن افراد مختلف بر روی یک پروژه نرم‌افزاری اتفاق بیفتد. در مقابل آن مثالی بیاورید که نشان می‌دهد چگونه مهندسانی که در کارهای مهندسی نرم‌افزاری مطلوب خبره هستند و از بررسی‌های فنی رسمی بهره می‌گیرند، می‌توانند نرخ تولید یک تیم را افزایش دهند (در مقایسه با حاصل جمع نرخ‌های تولید فردی). راهنمایی: می‌توانید فرض کنید که بررسی‌ها دوباره کاری را کاهش می‌دهد و دوباره کاری می‌تواند ۲۰ تا ۴۰ درصد زمان یک فرد را شامل باشد.

۵-۷ اگر چه اضافه کردن افراد به یک پروژه نرم‌افزاری به تأخیر افتاده آن را بیشتر به تأخیر می‌اندازد، اما مواقعی نیز چنین نمی‌شود. آنها را توضیح دهید.

۶-۷ رابطه بین افراد و زمان به میزان زیادی غیرخطی است. با به کار بردن معادله نرم‌افزاری پوتنام (که در بخش ۲-۲-۷ توصیف شد)، جدولی به وجود آورید که رابطه تعداد افراد به زمان پروژه را برای یک پروژه نرم‌افزاری نشان دهد. فرض کنید این پروژه به $LOC = 50,000$ و ۱۵ سال تلاش فردی نیاز دارد (پارامتر بازدهی $5/000$ است و $B = 0.37$). فرض کنید که نرم‌افزار با مثبت و منفی ۱۲ ماه در طول ۲۴ ماه باید تحویل گردد.

۷-۷ فرض کنید که شما برای توسعه یک سیستم ثبت مرتبط با یک دانشگاه قرارداد بسته‌اید. نخست مانند مشتری عمل کنید، (اگر دانشجو باشید، این کار باید برای شما آسان باشد) و خصوصیات یک سیستم خوب را تعیین کنید. [از طرف دیگر، مربی شما یک‌سری از نیازهای اولیه سیستم را به شما ارائه می‌دهد.] با به کار بردن روش‌های ارزیابی‌ای که در فصل ۵ بحث شدند، یک ارزیابی زمانی و کاری برای سیستم ثبت مرتبط این دانشگاه انجام دهید. بیان کنید که چگونه شما:

الف. فعالیت‌های کاری موازی را در مدت اجرای پروژه سیستم ثبت مرتبط تعریف خواهید کرد.

ب. تلاش را در سراسر پروژه توزیع خواهید کرد.

ج. هدفهای اساسی برای پروژه تعیین خواهید کرد.

۸-۷ با به‌کار بردن بخش ۳-۷ به‌عنوان یک راهنما TSS سیستم ثبت مرتبط دانشگاه را مناسبه کنید. از این‌که همه کارتان را نشان می‌دهید مطمئن شود. نوع پروژه و مجموعه کار مناسب برای پروژه را انتخاب کنید.

۹-۷ یک شبکه کار برای سیستم ثبت مرتبط دانشگاه، یا به‌جای آن، برای پروژه نرم‌افزاری دیگری که به آن علاقه‌مندید تعریف نمایید. از این‌که کارها و هدفهای اصلی و نیز برآوردهای زمانی و کاری مربوط به هر کار را نشان می‌دهید مطمئن شوید. اگر امکان دارد، برای انجام این کار یک ابزار زمان‌بندی خودکار به‌کار ببرید.

۱۰-۷ اگر یک ابزار زمان‌بندی خودکار در دسترس است، مسیر بحرانی را برای شبکه‌ای که در مسئله ۹-۷ تعریف شد تعیین کنید.

۱۱-۷ یک ابزار زمان‌بندی (اگر در دسترس است) یا کاغذ و مداد (اگر لازم است) به‌کار ببرید، و یک نمودار زمانی برای پروژه سیستم ثبت مرتبط دانشگاه تهیه نمایید.

۱۲-۷ کاری را که در بخش ۴-۷ خطرسنجی فناوری گفته شد، همانند تصحیح تعیین حوزه مفهوم در بخش ۵-۷ تصحیح کنید.

۱۳-۷ فرض کنید که شما یک مدیر پروژه نرم‌افزاری هستید و از شما خواسته شده است که آماری‌های ارزش اکتسابی یک پروژه نرم‌افزاری کوچک را محاسبه کنید. پروژه دارای ۵۶ کار برنامه‌ریزی شده است که تخمین زده شود نیاز به ۵۸۲ روز فردی برای کامل شدن دارد. در زمانی که از شما خواسته شده است تا ارزش‌سنجی اکتسابی را انجام دهید ۱۲ کار کامل شده‌اند. با این حال، زمان‌بندی پروژه نشان می‌دهد که ۱۵ کار باید کامل می‌شده‌اند.

اطلاعات زمان‌بندی ذیل (برحسب روزهای فردی) موجود می‌باشند:

نیروی کاری واقعی	نیروی کاری طرح ریزی شده	وظیفه
۱۲/۵	۱۲/۰	۱
۱۱/۰	۱۵/۰	۲
۱۷/۰	۱۳/۰	۳
۹/۵	۸/۰	۴
۹/۰	۹/۵	۵
۱۹/۰	۱۸/۰	۶
۱۰/۰	۱۰/۰	۷
۴/۵	۴/۰	۸
۱۰/۰	۱۲/۰	۹
۶/۵	۶/۰	۱۰

۱۱	۵/۰	۴/۰
۱۲	۱۴/۰	۱۴/۵
۱۳	۱۶/۰	—
۱۴	۶/۰	—
۱۵	۸/۰	—

شاخص اجرای برنامه (SPI)، واریانس برنامه، درصد برنامه‌ریزی شده برای تکمیل، درصد کامل،

شاخص اجرایی هزینه (SPI) و واریانس هزینه را برای پروژه محاسبه کنید.

۷-۱۴ آیا امکان دارد کارایی رفع نقص (DRE) را به عنوان متریکی برای ردیابی خطا در سراسر یک

پروژه نرم‌افزاری به کار برد؟ دلایل موافق و مخالف کاربرد کارایی رفع نقص برای این منظور را مورد بحث

قرار دهید.

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

فهرست منابع و مراجع

- [BR095] Brooks, M., *The Mythical Man-Month*, Anniversary Edition, Addison-Wesley, 1995.
- [FLE98] Fleming, Q.W. and J.M. Koppelman, "Earned Value Project Management," *Crosstalk*, vol. II, no. 7, July 1998, p. 19.
- [HUM95] Humphrey, W., *A Discipline for Software Engineering*, Addison-Wesley, 1995.
- [PAG85] Page-Jones, M., *Practical Project Management*, Dorset House, 1985, pp. 90-91.
- [PRE99] Pressman, R.S., *Adaptable Process Model*, R.S. Pressman & Associates, 1999.
- [PUT92] Putnam, L. and W. Myers, *Measures for Excellence*, Yourdon Press, 1992.
- [RIG81] Riggs, J., *Production Systems Planning, Analysis and Control*, 3rd ed., Wiley, 1981.
- [THE93] The', L., "Project Management Software That's IS Friendly," *Datamation*, October 1, 1993, pp. 55-58.
- [WIL99] Wilkens, T.T., "Earned Value, Clear and Simple," Primavera Systems, April 1, 1999, p. 2.
- [ZAH95] Zahniser, R., "Time-Boxing for Top Team Performance," *Software Development*, March 1995, pp. 34-38.

خواندنیهای دیگر و منابع اطلاعاتی

McConnell (*Rapid Development*, Microsoft Press, 1996) presents an excellent discussion of the issues that lead to overly optimistic software project scheduling and what you can do about it. O'Connell (*How to Run Successful Projects II: The Silver Bullet*, Prentice-Hall, 1997) presents a step-by-step approach to project management that will help you to develop a realistic schedule for your projects.

Project scheduling issues are covered in most books on software project management. McConnell (*Software Project Survival Guide*, Microsoft Press, 1998), Hoffman and Beaumont (*Application Development: Managing a Project's life Cycle*, Midrange Computing, 1997), Wysoki and his colleagues (*Effective Project Management*, Wiley, 1995), and Whitten (*Managing Software Development Projects*, 2nd ed., Wiley, 1995) consider the topic in detail. Boddie (*Crunch Mode*, Prentice-Hall, 1987) has written a book for all managers who "have 90 days to do a six month project."

Worthwhile information on project scheduling can also be obtained in general purpose project management books. Among the many offerings available are

Kerzner, H., *Project Management: A Systems Approach to Planning, Scheduling, and Control*, Wiley, 1998.

Lewis, J.P., *Mastering Project Management: Applying Advanced Concepts of Systems Thinking, Control and Evaluation*, McGraw-Hill, 1998.

Fleming and Koppelman (*Earned Value Project Management*, Project Management Institute Publications, 1996) discuss the use of earned value techniques for project tracking and control in considerable detail.

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

<http://www.mhhe.com/engcs/compscipressman/resources/project-sched.mhtml>