Project Management

اهداف

- اهمیت سیستم‌های اطلاعاتی در کسب‌وکار را به خوبی درک کنید.
- توانایی ایجاد درخواست سیستم را داشته باشید.
- نحوه امکان‌سنجی فنی، اقتصادی و سازمانی را فرا بگیرید.
- بتوانید تحلیل امکان‌سنجی را به درستی انجام دهید.
- با فرآیند انتخاب پروژه‌ها در برخی سازمان‌ها آشنا شوید.
- با ابزارهای مدیریتی مانند ساختار شکست کار(WBS) ، نمودار گانت و نمودار شبکه‌ای آشنا شوید.
- با روش‌های تخمین تلاش و زمان پروژه بر اساس Use Case آشنایی پیدا کنید.
- بتوانید یک برنامه کاری تکرارپذیر برای پروژه تهیه کنید.
- نحوه مدیریت محدوده پروژه، اصلاح تخمین‌ها و مدیریت ریسک را یاد بگیرید.
- چگونگی جذب و سازماندهی نیروی انسانی برای پروژه را بدانید.
- بفهمید که چگونه محیط و زیرساخت‌های یک سازمان با فرآیند مدیریت پروژه تعامل دارند.

انتخاب پروژه

یک نیاز جدید برای بهبود کسب‌وکار وجود دارد و برای رفع آن باید سیستمی جدید ایجاد شود. این نیاز ممکن است:

- از بخش‌های مختلف سازمان، مانند واحد کسب‌وکار یا فناوری اطلاعات و ... مطرح و پیشنهاد شود.
  به‌عنوان مثال، برای جذب مشتریان جدید یا بهبود همکاری با تأمین‌کنندگان سیستم جدید طراحی شود.

- نیاز به پروژه ناشی از مشکلاتی درون سازمان است، مثل کاهش فروش یا خدمات ضعیف به مشتریان.

- سازمان‌ها به دنبال استفاده از فناوری‌های جدید و نوآورانه هستند تا از رقبا جلوتر باشند و بتوانند اولین عرضه‌کننده یک محصول یا خدمات جدید باشند.

انتخاب پروژه

حامی پروژه (**Project Sponsor**) فردی است که این نیاز را درک می‌کند و به موفقیت آن پروژه علاقه‌مند است. او کسی است که به تیم پروژه کمک می‌کند تا مطمئن شوند کارها درست انجام می‌شود. حامی مدیران ارشد مختلف یک سازمان نیز می تواند باشد گاهی هم فناوری اطلاعات ممکن است نقش حامی را داشته باشد.

اندازه و اهمیت پروژه تعیین می‌کند که حامی چه ویژگی ها و امکاناتی باید داشته باشد. پروژه‌های کوچک ممکن است به حمایت یک مدیر میانی نیاز داشته باشند، در حالی که پروژه‌های بزرگ و پیچیده‌تر ممکن است نیاز به حمایت مدیریت ارشد یا حتی مدیرعامل شرکت داشته باشند.

پس از شناسایی نیاز کسب‌وکار و تعیین اینکه سیستم جدید چه کاری باید انجام دهد و همچنین مشخص شدن حامی، حامی پروژه ارزش‌هایی که سیستم جدید برای سازمان ایجاد خواهد کرد را مدون می کند. این ارزش‌ها می‌توانند ملموس باشند (مثل کاهش هزینه‌ها) یا غیرملموس (مثل بهبود خدمات مشتری).

وقتی این موارد مدون شد، حامی پروژه به‌طور رسمی درخواست راه‌اندازی پروژه را ثبت می‌کند که به آن "درخواست سیستم" می‌گویند.

System Request

درخواست سیستم، یک سند است، که دلایل ساخت یک سیستم و ارزش آن برای کسب‌وکار را توضیح می‌دهد. این سند توسط حامی پروژه تهیه شده و شامل **پنج بخش** است:

حامی پروژه (Project Sponsor): فرد مسئول پیگیری پروژه.

نیاز کسب‌وکار: دلیل اصلی نیاز به سیستم.

الزامات کسب‌وکار: قابلیت‌هایی که سیستم باید داشته باشد.

ارزش کسب‌وکار: مزایای پیش‌بینی‌شده برای سازمان (مثل کاهش هزینه یا بهبود خدمات).

مسائل ویژه: شرایط خاصی که باید در نظر گرفته شوند، مثل مهلت‌های خاص.

این درخواست به کمیته تأیید ارسال می‌شود که تصمیم می‌گیرد آیا پروژه بررسی بیشتری نیاز دارد یا نه. اگر تأیید شد، مرحله بعدی تحلیل امکان‌سنجی است.

System Request

System Request—Name of Project
حامی پروژه : نام حامی پروژه
نیاز کسب و کار: شرح مختصری از نیاز کسب و کار
الزامات کسب کار: شرح الزامات تجاری
ارزش کسب و کار: ارزش مورد انتظاری که سیستم ارائه خواهد کرد
مسائل ویژه یا محدودیت های خاص: هر گونه اطلاعات اضافی که ممکن است به ذینفعان مربوط باشد

حامی پروژه:

عباس رضائی

مدیر دفتر سیستم ها و فناوری اطلاعات

نیاز کسب‌وکار:

شرکت برای بهبود فرآیند ثبت سفارشات و کاهش خطاهای انسانی به یک سیستم خودکار ثبت سفارش نیاز دارد. در حال حاضر، ثبت سفارشات به صورت دستی انجام می‌شود که منجر به تأخیر در پردازش و افزایش احتمال خطا می‌گردد.

الزامات کسب‌وکار:

قابلیت ثبت و پیگیری سفارشات به صورت آنلاین.

امکان صدور فاکتور به صورت خودکار.

گزارش‌گیری جامع از وضعیت سفارشات.

یکپارچگی با سیستم‌های موجود (CRMو انبار).

ارزش کسب‌وکار:

صرفه جویی در زمان : کاهش زمان پردازش سفارشات و خطاهای انسانی.

بهبود خدمات: افزایش سرعت پاسخگویی به مشتریان و کاهش زمان انتظار.

افزایش رضایت مشتری: ارائه خدمات سریع‌تر و دقیق‌تر.

مسائل ویژه:

مهلت نهایی برای پیاده‌سازی سیستم، سه ماه بعد از تأیید درخواست.

نیاز به آموزش کارکنان برای استفاده از سیستم جدید.

Feasibility Analysis

تحلیل امکان سنجی

پس از این که نیازهای سیستم و الزامات کسب‌وکار مشخص شد، باید یک **طرح تجاری** دقیق‌تر تهیه شود تا فرصت‌ها و محدودیت‌های پروژه پیشنهادی بهتر شناخته شود. تحلیل قابلیت انجام، به سازمان کمک می‌کند تا تصمیم بگیرد که:

- آیا باید پروژه را ادامه دهد یا خیر. این تحلیل همچنین خطرات مهمی که با پروژه همراه است را شناسایی می‌کند و نشان می‌دهد که اگر پروژه برای انجام تأیید شود، باید این خطرات نیز در نظر گرفته شوند.

هر سازمان روش و فرمت خاص خود را برای تحلیل قابلیت انجام دارد، اما معمولاً شامل سه نوع تحلیل است:

- **امکان سنجی فنی**
- **امکان سنجی اقتصادی**
- **امکان سنجی سازمانی**

نتایج این تحلیل‌ها به کمیته تأیید ارائه می‌گردد.

اگر در هر زمان خطرات و محدودیت‌های پروژه بیشتر از مزایای آن باشد، تیم پروژه ممکن است تصمیم بگیرد که پروژه را لغو کند یا تغییرات لازم را انجام دهد.

امکان سنجی فنی

قابلیت فنی: آیا می‌توانیم آن را بسازیم؟

تحلیل قابلیت فنی پروژه به این سوال پاسخ می‌دهد که آیا می‌توانیم سیستم مورد نظر را بسازیم. این تحلیل به شناسایی ریسک‌های فنی کمک می‌کند که ممکن است بر موفقیت پروژه تأثیر بگذارد.

عوامل مؤثر بر ریسک های فنی

عدم آشنایی با حوزه کاری: اگر کاربران و تحلیل‌گران با حوزه کسب‌وکار آشنا نباشند، احتمال سوءتفاهم و از دست دادن فرصت‌های بهبود افزایش می‌یابد. توسعه سیستم‌های جدید معمولاً خطر بیشتری نسبت به اصلاح سیستم‌های موجود دارد.

فناوری های جدید: استفاده از فناوری‌های جدید در سازمان می‌تواند منجر به بروز مشکلات و تأخیرها شود، به‌خصوص اگر تیم توسعه با آن فناوری آشنا نباشد.

اندازه پروژه: پروژه‌های بزرگ‌تر معمولاً ریسک بیشتری دارند زیرا مدیریت آن‌ها پیچیده‌تر است و ممکن است الزامات مهم نادیده گرفته شوند.

سازگاری: سیستم‌های جدید باید با فناوری‌های موجود سازگار باشند، زیرا ممکن است به داده‌ها و زیرساخت‌های موجود وابسته باشند.

ارزیابی قابلیت فنی

ارزیابی قابلیت فنی نیاز به بررسی شرایط و مقایسه پروژه با پروژه‌های قبلی سازمان دارد. همچنین مشاوره با کارشناسان IT می‌تواند در ارزیابی قابلیت فنی پروژه کمک‌کننده باشد.

امکان سنجی مالی

قابلیت اقتصادی: آیا باید آن را بسازیم؟

دومین عنصر تحلیل قابلیت انجام، تحلیل قابلیت اقتصادی (که به آن تحلیل هزینه-فایده نیز گفته می‌شود) است که به شناسایی ریسک‌های مالی مرتبط با پروژه می‌پردازد. این تحلیل به این سوال پاسخ می‌دهد: آیا باید سیستم را بسازیم؟

مراحل انجام تحلیل هزینه-فایده به شرح زیر است :

- شناسایی هزینه‌ها و مزایا
- تخصیص مقادیر به هزینه‌ها و مزایا
- محاسبه جریان نقدی
- ارزیابی بازگشت سرمایه

این مراحل به تصمیم‌گیری صحیح در مورد ادامه یا عدم ادامه پروژه کمک می‌کند.

امکان سنجی مالی

شناسایی هزینه ها و مزایا	هزینه ها و منافع ملموس پروژه را فهرست کنید. شامل هزینه های یکبار مصرف و مکرر می شود.
تخصیص مقادیر به هزینه‌ها و مزایا	برای هر هزینه و مزایا، مقادیر عددی تعیین شود تا بتوان ارزیابی دقیق‌تری از جنبه‌های اقتصادی پروژه داشت. حتی هزینه‌ها و مزایای نامشهود، که ممکن است به‌راحتی قابل اندازه‌گیری نباشند (مانند افزایش رضایت مشتری یا بهبود شهرت شرکت)
محاسبه جریان نقدی	با تحلیل هزینه‌ها و مزایای احتمالی برای سه تا پنج سال آینده، می‌توان دید بهتری از تأثیرات مالی پروژه به‌دست آورد. همچنین، اگر انتظار می‌رود که هزینه‌ها یا درآمدها در آینده افزایش یابد، می‌توان از نرخ رشد برای محاسبه مقادیر استفاده کرد.
تعیین ارزش خالص فعلی (NPV) :	این فرایند به شما کمک می‌کند تا بدانید هزینه‌ها و مزایای آینده چه ارزشی خواهند داشت اگر به معیارهای مالی امروز تطبیق داده شوند. برای این کار، لازم است یک نرخ رشد انتخاب کنید که در فرمول NPVبه کار می‌رود. این محاسبه به تصمیم‌گیری در مورد توجیه اقتصادی پروژه کمک می‌کند.

امکان سنجی مالی

تعیین بازگشت سرمایه (ROI):	محاسبه کنید که سازمان چه مقدار پول به‌ازای سرمایه‌گذاری که انجام می‌دهد، بدست خواهد آورد، با استفاده از فرمول ROI
تعیین نقطه سربه‌سر	یعنی مزایای مالی پروژه از هزینه‌های آن بیشتر شود. با استفاده از فرمول نقطه سربه‌سر، می‌توانید محاسبه کنید که چه مدت طول می‌کشد تا پروژه به مرحله‌ای برسد که بازگشت هزینه‌ها را تضمین کند و شروع به ایجاد ارزش واقعی برای سازمان کند.
رسم نمودار نقطه سربه‌سر	در این روش، هزینه‌ها و مزایای سالانه به‌صورت خطوط روی نمودار رسم می‌شوند. جایی که این دو خط یکدیگر را قطع کنند، همان نقطه سربه‌سر است، یعنی لحظه‌ای که مزایا با هزینه‌ها برابر شده و پروژه شروع به سودآوری می‌کند.

امکان سنجی سازمانی

تحلیل قابلیت سازمانی به بررسی اینکه آیا سیستم جدید به خوبی توسط کاربران پذیرفته می‌شود و چگونه در عملیات جاری سازمان ادغام خواهد شد، می‌پردازد هدف آن پاسخ به این سؤال است: "اگر ما آن را بسازیم، آیا آنها استفاده خواهند کرد؟".

روش‌های امکان سنجی سازمانی :

- **هم‌راستایی استراتژیک**:

یکی از راه‌های ارزیابی قابلیت سازمانی، بررسی هم‌راستایی اهداف پروژه با اهداف کسب‌وکار است. هر چه هم‌راستایی بیشتر باشد، پروژه از نظر قابلیت سازمانی کمتر ریسک خواهد داشت. به عنوان مثال، اگر دپارتمان بازاریابی بخواهد بر مشتری‌مداری تمرکز کند، یک پروژه CRMکه اطلاعات یکپارچه مشتری را تولید کند، با اهداف بازاریابی هم‌راستا خواهد بود.

- **تحلیل ذینفعان**:

ذینفعان افرادی هستند که می‌توانند تحت تأثیر سیستم جدید قرار گیرند یا بر آن تأثیر بگذارند.تحلیل دقیق ذینفعان به درک نیازها و انتظارات آنها کمک می‌کند. برای مثال، دپارتمان IS (information system) می‌تواند به عنوان یک ذینفع در نظر گرفته شود، زیرا نقش‌ها یا مشاغل آن ممکن است بعد از پیاده‌سازی سیستم به‌طور قابل توجهی تغییر کند.

- **حامی پروژه:**

حامی پروژه ممکن است یک مقام غیر IT با سطح بالا است که معمولاً اسپانسر پروژه است و سیستم را درخواست کرده است. این فرد با اختصاص زمان، منابع و حمایت های دیگر به پروژه کمک می‌کند. وجود چند حامی بهتر است تا در صورت خروج یکی از آنها، حمایت پروژه از بین نرود.

مشارکت کاربران باید در کل فرایند توسعه وجود داشته باشد.

انتخاب پروژه

پس از تکمیل امکان سنجی، این تحلیل به کمیته تأیید ارائه می‌شود، کمیته تصمیم می‌گیرد که آیا پروژه را تأیید کند، رد کند. کمیته تایید پروژه را با بررسی نیاز کسب‌وکار (موجود در درخواست سیستم) و ریسک‌های ساخت سیستم (موجود در امکان سنجی) مورد ارزیابی قرار می‌دهد.

برای تأیید پروژه، کمیته همچنین پروژه را از منظر سازمانی بررسی می‌کند و باید کل پرتفوی پروژه‌های شرکت را در نظر بگیرد. این نوع مدیریت پروژه به‌عنوان **مدیریت پرتفوی** شناخته می‌شود. مدیریت پرتفوی به انواع مختلف پروژه‌هایی که در یک سازمان وجود دارد پروژه‌های بزرگ و کوچک، با ریسک بالا و پایین، استراتژیک و تاکتیکی توجه می‌کند. یک پرتفوی پروژه خوب، مناسب‌ترین ترکیب پروژه‌ها برای نیازهای سازمان را دارد.

کمیته به‌عنوان مدیر پرتفوی عمل می‌کند با هدف به حداکثر رساندن عملکرد هزینه-فایده و سایر عوامل مهم پروژه‌ها در پرتفوی خود.

اگر سه پروژه بالقوه با بازده بالا وجود داشته باشد، اما همه آنها ریسک بسیار بالایی داشته باشند، ممکن است تنها یکی از پروژه‌ها انتخاب شود.

ابزارهای مدیریت پروژه به روش سنتی:

ابتدا نیاز است که مجموعه‌ای از ابزارهای مدیریت پروژه که برای مدیریت موفق پروژه‌های توسعه نرم‌افزار (و بسیاری دیگر از انواع پروژه‌ها) استفاده شده‌اند، معرفی کنیم. این ابزارها شامل:

- **ساختار شکست کار(Work Breakdown Structures)**
- **نمودار گانت**
- **دیاگرام شبکه**

مراحل مدیریت پروژه

شناسایی وظایف: اولین کاری که یک مدیر پروژه باید انجام دهد، شناسایی وظایفی است که باید انجام شوند و تعیین زمان لازم برای هر وظیفه است.

سازماندهی وظایف: پس از شناسایی و مستند کردن وظایف، آن‌ها در یک **ساختار شکست کار** سازماندهی می‌شوند. این ساختار به عنوان مبنایی برای ایجاد نمودارهای گانت استفاده می‌شود.

ایجاد نمودارها: نمودارهای گانت و دیاگرام شبکه برای نمایش گرافیکی workflow استفاده می‌شوند. این تکنیک‌ها به مدیر پروژه کمک می‌کنند تا پیشرفت پروژه را در طول زمان درک و مدیریت کند.

این ابزارها، در کنار شناسایی و مستند کردن وظایف، اساس کار مدیریت پروژه را تشکیل می‌دهند و به مدیران پروژه کمک می‌کنند تا بر روند پیشرفت پروژه نظارت داشته باشند و آن را کنترل کنند.

Work Breakdown Structures(WBS)

تعریف وظیفه یا Task

یک وظیفه، واحدی از کار است که توسط یک یا گروهی از اعضای تیم توسعه انجام می‌شود. به عنوان مثال، تحلیل قابلیت می‌تواند یک وظیفه باشد.

هر وظیفه معمولا شامل:

- نام وظیفه
- تاریخ شروع و پایان
- فرد منصوب برای انجام وظیفه
- تحویل‌دهنده‌ها
- وضعیت تکمیل
- اولویت
- منابع مورد نیاز
- زمان تخمینی برای تکمیل وظیفه
- و زمان واقعی که برای تکمیل آن صرف شده است

Work Breakdown Structures(WBS)

مدیر پروژه می‌تواند از یک رویکرد ساختاریافته و از بالا به پایین استفاده کند، به این صورت که ابتدا وظایف سطح بالا تعریف می‌شوند و سپس به زیر وظایف شکسته می‌شوند.

مراحل WBS

- تعریف وظایف سطح بالا

برای مثال، در زیر، فهرستی از وظایف سطح بالا برای اجرای یک کلاس آموزشی جدید در زمینه فناوری اطلاعات نشان داده شده است. مراحل اصلی این فرآیند شامل موارد زیر است :

1. **شناسایی تأمین‌کنندگان**
2. **بررسی مطالب آموزشی**
3. **مقایسه تأمین‌کنندگان**
4. **مذاکره با تأمین‌کنندگان**
5. **توسعه کانال های ارتباطی**
6. **انتشار اطلاعات**
7. **ایجاد و مدیریت نظر سنجی**
8. **تحلیل نتایج و انتخاب فروشنده**
9. **ساخت کلاس‌های جدید**
10. **توسعه گزینه‌های دوره آموزشی**

Task Number	Task Name	Duration (in weeks)	Dependency	Status
1	Identify vendors	2		Complete
2	Review training materials	6	1	Complete
3	Compare vendors	2	2	In Progress
4	Negotiate with vendors	3	3	Open
5	Develop communications information	4	1	In Progress
6	Disseminate information	2	5	Open
7	Create and administer survey	4	6	Open
7.1	Create initial survey	1		Open
7.2	Review initial survey	1	7.1	Open
7.2.1	Review by Director of IT Training	1		Open
7.2.2	Review by Project Sponsor	1		Open
7.2.3	Review by Representative Trainee	1		Open
7.3	Pilot test initial survey	1	7.1	Open
7.4	Incorporate survey changes	1	7.2, 7.3	Open
7.5	Create distribution list	0.5		Open
7.6	Send survey to distribution list	0.5	7.4, 7.5	Open
7.7	Send follow-up message	0.5	7.6	Open
7.8	Collect completed surveys	1	7.6	Open
8	Analyze results and choose vendor	2	4, 7	Open
9	Build new classrooms	11	1	In Progress
10	Develop course options	3	8, 9	Open

شکستن وظایف به زیر وظایف:

هر مرحله به زیر مراحل شکسته می‌شود و به‌صورت سلسله‌مراتبی شماره‌گذاری می‌گردد. به‌عنوان مثال، برای **ایجاد و مدیریت نظرسنجی(۷)**، هشت زیر وظیفه وجود دارد (یعنی ۷.۱ تا ۷.۸) و سه زیر وظیفه برای ۷.۲ یعنی (۷.۲.۱ تا ۷.۲.۳) که شامل بررسی نظرسنجی اولیه است.

- **ایجاد نظرسنجی اولیه**
- **بررسی نظرسنجی اولیه**
  - **بررسی توسط مدیر آموزش فناوری اطلاعات**
  - **بررسی توسط حامی پروژه**
  - **بررسی توسط نماینده کارآموز**
- **آزمایش پایلوت نظرسنجی اولیه**
- **انجام تغییرات در نظرسنجی**
- **ایجاد فهرست توزیع**
- **ارسال نظرسنجی به فهرست توزیع**
- **ارسال پیام پیگیری**
- **جمع‌آوری نظرسنجی‌های تکمیل شده**

ساختار شکست کار:

WBS باید شامل موارد زیر باشد:

- **مدت زمان وظیفه**
- **وضعیت فعلی وظیفه** (یعنی باز یا تمام شده)
- **وابستگی‌های وظیفه** یعنی یک وظیفه نمی‌تواند انجام شود تا زمانی که یک وظیفه دیگر به پایان برسد.

به‌عنوان مثال، در شکل سلاید قبل نشان داده شده است که انجام تغییرات در نظرسنجی (وظیفه ۷.۴) یک هفته طول می‌کشد، اما نمی‌تواند تا بعد از بررسی نظرسنجی (وظیفه ۷.۲) و آزمایش اولیه (وظیفه ۷.۳) انجام شود.

شناسایی نقاط عطف: می بایست نقاط عطف کلیدی یا تاریخ‌های مهم نیز در برنامه کاری شناسایی می‌شوند.

رویکردهای سازماندهی WBS

دو رویکرد اصلی برای سازماندهی یک WBS وجود دارد:

بر اساس مراحل توسعه:

فرض کنید یک شرکت تصمیم بگیرد که یک وب‌سایت توسعه دهد، می‌تواند یک WBS بر اساس مراحل:

- شروع (inception)
- توسعه (elaboration)
- ساخت (construction)
- انتقال (transition)

ایجاد کند. در این حالت، یک وظیفه معمولی که در مرحله شروع اتفاق می‌افتد، فرضا می تواند امکان سنجی باشد که به انواع مختلف (فنی، اقتصادی و سازمانی) شکسته می‌شود.

بر اساس محصولات:

شرکت می‌تواند برنامه کاری را بر اساس محصولات مختلفی که قرار است توسعه یابند، سازماندهی کند.

فرضا در مورد وب سایت فوق، محصولات می‌توانند شامل اپلت‌ها(برنامه های کوچکتر ذیل برنامه های اصلی مثل یک برنامه فلش بر روی صفحه وب)، سرورهای برنامه، سرورهای پایگاه داده، مجموعه‌های مختلف صفحات وب، نقشه سایت و غیره باشند. سپس این موارد به وظایف مختلف مرتبط با مراحل فرآیند توسعه شکسته می‌شوند.

نمودار گانت

نمودار گانت یک نمودار میله‌ای افقی است که همان اطلاعات مربوط به وظایف را که در ساختار شکست کار WBSپروژه موجود است، به صورت گرافیکی نمایش می‌دهد. گاهی اوقات یک تصویر واقعاً از هزار کلمه ارزشمندتر است و نمودار گانت می‌تواند وضعیت کلی پروژه را بسیار سریع‌تر و آسان‌تر از WBSمنتقل کند. ایجاد نمودار گانت ساده است و می‌توان آن را با استفاده از نرم‌افزارهای صفحه‌گسترده، نرم‌افزارهای گرافیکی یا نرم‌افزارهای مدیریت پروژه تهیه کرد.

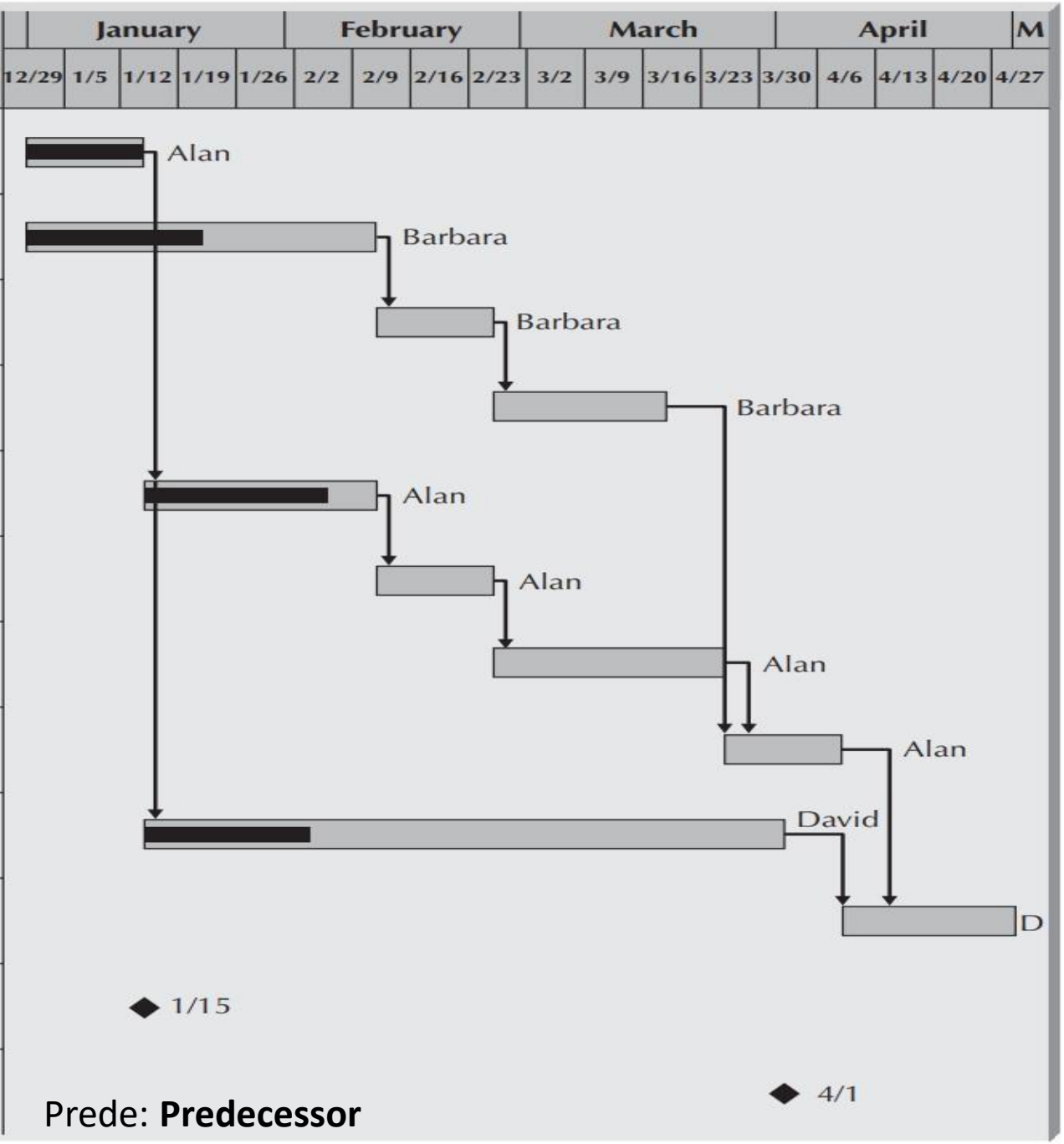
ابتدا وظایف به صورت ردیف‌ها در نمودار فهرست می‌شوند و زمان به صورت دوره‌های زمانی بر روی قسمت بالایی نمودار نمایش داده می‌شود که بر اساس نیازهای پروژه تعیین می‌شود. برای پروژه‌های کوتاه، زمان ممکن است به ساعت یا روز تقسیم شود، در حالی که برای پروژه‌های متوسط، زمان به هفته یا ماه تقسیم می‌شود.

میله‌های افقی برای نشان دادن مدت زمان هر وظیفه ترسیم می‌شوند؛ ابتدای هر میله و انتهای آن دقیقاً نشان می‌دهد که وظیفه کی آغاز و کی به پایان می‌رسد. هنگامی که افراد بر روی وظایف کار می‌کنند، میله‌های مربوطه به نسبت مقداری که از وظیفه انجام شده، پر می‌شوند. اگر تعداد وظایف در یک نمودار گانت زیاد باشد، می‌تواند باعث سردرگمی شود، بنابراین بهتر است تعداد وظایف را به حدود بیست یا سی محدود کنید. اگر وظایف بیشتری وجود داشته باشد، آنها را به زیر وظایف تقسیم کرده و برای هر سطح جزئیات نمودار گانت جداگانه ایجاد کنید.

مدیر پروژه می‌تواند با نگاه کردن به نمودار گانت به سرعت بسیاری از نکات را متوجه شود. علاوه بر دیدن اینکه وظایف چه مدت طول می‌کشند و چه مقدار پیشرفت کرده‌اند، مدیر پروژه می‌تواند ببیند که کدام وظایف پیوسته هستند، کدام وظایف همزمان انجام می‌شوند و کدام وظایف به نوعی همپوشانی دارند. او می‌تواند با کشیدن یک خط عمودی بر روی تاریخ امروز، نگاهی سریع به وظایفی که از زمان‌بندی جلوتر یا عقب‌تر هستند داشته باشد. اگر میله‌ای پر نشده و در سمت چپ خط باشد، به این معنی است که آن وظیفه از زمان‌بندی عقب است.

چندین علامت ویژه نیز می‌توان بر روی نمودار گانت قرار داد. نقاط عطف پروژه با مثلث‌های وارونه یا الماس نشان داده می‌شوند. پیکان‌هایی بین میله‌های وظیفه برای نشان دادن وابستگی‌های وظیفه کشیده می‌شوند. گاهی اوقات، نام‌های افرادی که به هر وظیفه اختصاص داده شده‌اند در کنار میله‌های وظیفه برای نشان دادن منابع انسانی تخصیص یافته به وظایف فهرست می‌شود.

ID	Task Name	Duration	Start	Finish	Prede
1	Identify vendors	2 wks	Wed 1/1/15	Tue 1/14/15	
2	Review training materials	6 wks	Wed 1/1/15	Tue 2/11/15	
3	Compare vendors	2 wks	Wed 2/12/15	Tue 2/25/15	2
4	Negotiate with vendors	3 wks	Wed 2/26/15	Tue 3/8/15	3
5	Develop communications information	4 wks	Wed 1/15/15	Tue 2/11/15	1
6	Disseminate information	2 wks	Wed 2/12/15	Tue 2/25/15	5
7	Create and administer survey	4 wks	Wed 2/26/15	Tue 3/25/15	6
8	Analyze results and choose	2 wks	Wed 3/26/15	Tue 4/8/15	4, 7
9	Build new classroom	11 wks	Wed 1/15/15	Tue 4/1/15	1
10	Develop course options	3 wks	Wed 4/9/15	Tue 4/29/15	8, 9
11	Budget Meeting	1 day	Wed 1/15/15	Wed 1/15/15	
12	Software Installation	1 day	Tue 4/1/15	Tue 4/1/15	



Prede: **Predecessor**

Network Diagram

دیاگرام شبکه یک روش گرافیکی دیگر برای نمایش اطلاعات برنامه‌ریزی پروژه است که وظایف پروژه را به‌صورت یک فلوچارت ترسیم می‌کند.

تکنیک ارزیابی و بررسی برنامه یا (Program Evaluation and Review Technique)PERT :

یک تکنیک تحلیل شبکه است که می‌تواند زمانی استفاده شود که **برآورد زمان برای هر وظیفه به‌طور نسبی نامشخص باشد.**
به‌جای استفاده از یک برآورد، برای زمان، از سه برآورد زمانی استفاده می‌کند: خوشبینانه، محتمل‌ترین و بدبینانه. سپس این سه برآورد را با استفاده از فرمول زیر به یک برآورد میانگین وزنی ترکیب می‌کند:

O = برآورد خوشبینانه(optimistic estimate)

M = برآورد محتمل‌ترین(most likely estimate)

P = برآورد بدبینانه(pessimistic estimate)

$$\text{زمان برآورد شده} = \frac{(O + 4M + P)}{6}$$

مثال:

فرض کنید زمان خوش‌بینانه برای تکمیل یک فعالیت ۴ روز، زمان محتمل ۶ روز و زمان بدبینانه ۱۲ روز است. با استفاده از فرمول، زمان مورد انتظار برای تکمیل این فعالیت :

$$6.67 \approx \frac{40}{6} = \frac{12 + 24 + 4}{6} = \frac{12 + (6)4 + 4}{6}$$

گره‌ها یاNodes : هر گره یک وظیفه را نشان می‌دهد. هر گره شامل اطلاعاتی مانند نام وظیفه، زمان تخمینی برای انجام آن و وضعیت فعلی است (نیمه تمام یا کامل).

قوس‌ها یا Arcs : خطوطی که دو گره را به هم متصل می‌کنند، نشان‌دهنده وابستگی بین دو وظیفه هستند. این یعنی یک وظیفه باید قبل از شروع وظیفه دیگر کامل شود.

ویژگی‌های نمودار شبکه:

نمودارهای نیمه‌کامل معمولاً با یک خط قطری درون گره نمایش داده می‌شوند و وظایف کامل شده با خطوط متقاطع مشخص می‌شوند.

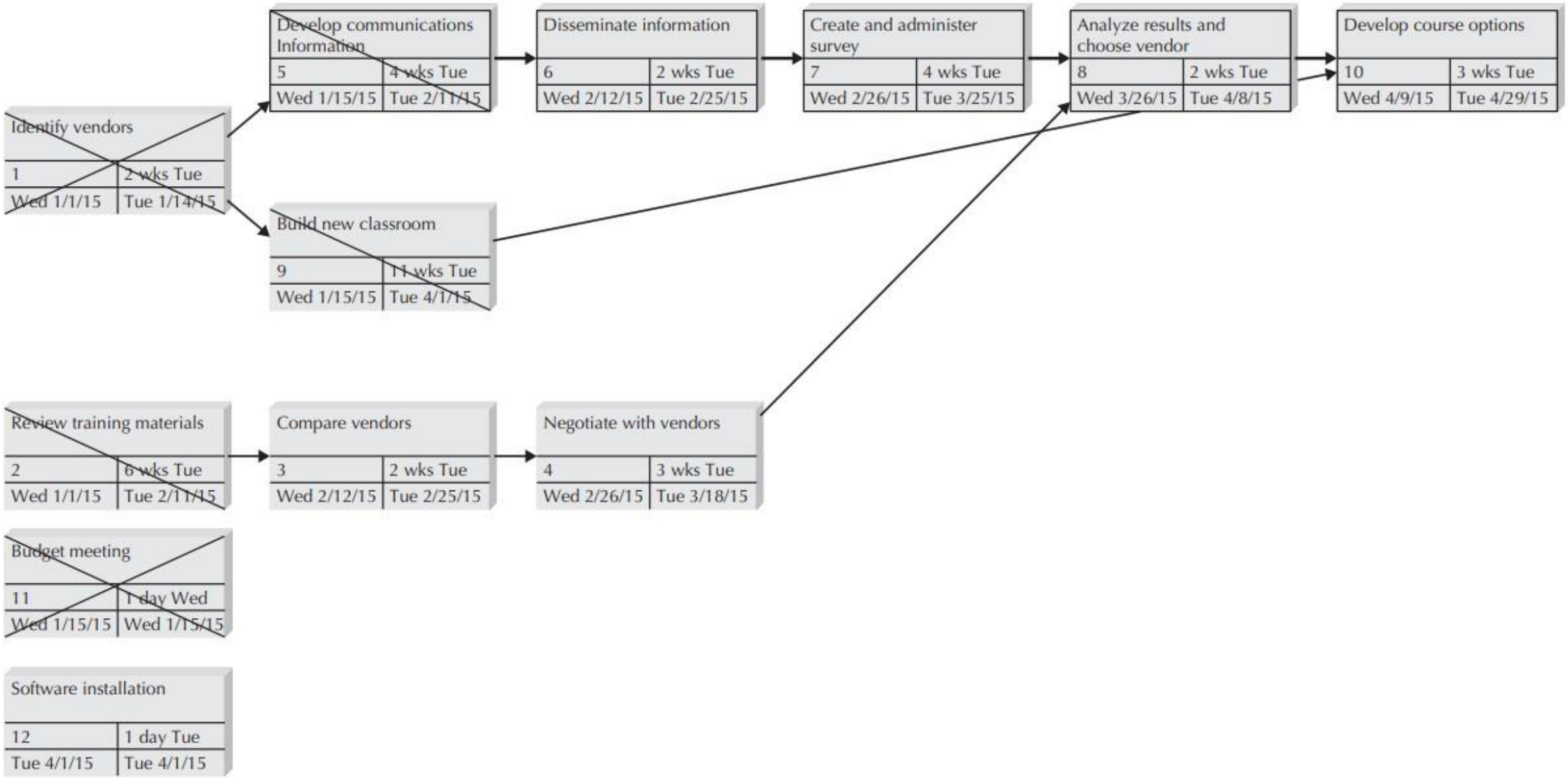
نمودارهای شبکه بهترین روش برای ارتباط وابستگی‌های وظیفه‌ای هستند زیرا وظایف را به ترتیبی که باید تکمیل شوند، نمایش می‌دهند.

روش مسیر بحرانی (CPM(The critical path method

روش مسیر بحرانی به سادگی اجازه شناسایی مسیر بحرانی در شبکه را می‌دهد. مسیر بحرانی، طولانی‌ترین مسیر از آغاز پروژه تا اتمام آن است. این مسیر تمامی وظایفی را که باید به‌موقع تکمیل شوند تا پروژه به‌طور کلی به‌موقع به پایان برسد، نشان می‌دهد.

اگر هر یک از وظایف در مسیر بحرانی بیشتر از حد انتظار طول بکشد، کل پروژه به تأخیر خواهد افتاد. هر وظیفه‌ای در مسیر بحرانی یک وظیفه بحرانی است و معمولاً به‌طور منحصر به‌فرد نمایش داده می‌شود؛ در شکل زیر این وظایف با حاشیه‌های دوتایی نشان داده شده‌اند (به وظایف ۵، ۶، ۷، ۸ و ۱۰ مراجعه کنید).

CPMمی‌تواند همراه با PERTیا بدون آن استفاده شود.


Identify vendors
1
2 wks Tue
Wed 1/1/15
Tue 1/14/15
Develop communications Information
5
4 wks Tue
Wed 1/15/15
Tue 2/11/15
Disseminate information
6
2 wks Tue
Wed 2/12/15
Tue 2/25/15
Create and administer survey
7
4 wks Tue
Wed 2/26/15
Tue 3/25/15
Analyze results and choose vendor
8
2 wks Tue
Wed 3/26/15
Tue 4/8/15
Develop course options
10
3 wks Tue
Wed 4/9/15
Tue 4/29/15
Build new classroom
9
11 wks Tue
Wed 1/15/15
Tue 4/1/15
Review training materials
2
6 wks Tue
Wed 1/1/15
Tue 2/11/15
Compare vendors
3
2 wks Tue
Wed 2/12/15
Tue 2/25/15
Negotiate with vendors
4
3 wks Tue
Wed 2/26/15
Tue 3/18/15
Budget meeting
11
1 day Wed
Wed 1/15/15
Wed 1/15/15
Software installation
12
1 day Tue
Tue 4/1/15
Tue 4/1/15

برآورد زمان لازم برای توسعه یک نرم افزار

علم (یا هنر) مدیریت پروژه در ایجاد تعادل بین سه مفهوم کلیدی است:

- **کارایی سیستم**
- **زمان اتمام پروژه**
- **هزینه پروژه**

این سه مفهوم را می‌توان به عنوان عوامل وابسته به هم در نظر گرفت که مدیر پروژه در طول توسعه سیستم کنترل می‌کند. هر بار که یکی از این عوامل تغییر کند، دو عامل دیگر نیز تحت تأثیر قرار می‌گیرند.

به عنوان مثال:

اگر مدیر پروژه نیاز داشته باشد که پروژه را سریعتر به اتمام برساند (یعنی عامل زمان را تغییر دهد) ، راه‌حل، کاهش کارایی سیستم یا افزایش هزینه‌ها از طریق اضافه کردن افراد بیشتر یا کار اضافی آنهاست.

مدیر پروژه غالباً با حامی پروژه در تعامل است تا اهداف پروژه را بروز نگه دارد ، مانند توسعه سیستمی با کارایی کمتر یا تمدید مهلت نهایی. در ابتدای پروژه، مدیر باید هر یک از این اهرم‌ها را برآورد کند و سپس به طور مداوم ارزیابی کند که چگونه پروژه را به گونه‌ای پیش ببرد که نیازهای سازمان را برآورده کند.

برآورد زمان لازم برای توسعه یک نرم افزار

روش‌های مختلفی برای برای برآورد زمان لازم برای توسعه یک سیستم وجود دارد. در اینجا چون فرآیند ها مبتنی بر use-case ها هستند، ما از روش use-case points استفاده می کنیم. یعنی سراغ محاسبه UCP ها می رویم.

Use-case به توصیف نحوه تعامل بازیگران (Actor) با یک سیستم برای رسیدن به یک هدف خاص کمک می‌کند

use-case models دارای دو ساختار اصلی هستند: actors و use cases

Actor :

نمایانگر نقشی است که کاربر سیستم ایفا می‌کند، نه یک کاربر خاص. به عنوان مثال، یک نقش می‌تواند منشی یا مدیر باشد. برای برآورد use-case points، actors می‌توانند به سه دسته زیر تقسیم می شوند:

- **ساده**: سیستم‌های جداگانه‌ای هستند که سیستم جاری باید از طریق یک رابط برنامه‌نویسی کاربردی (API) مشخص با آن‌ها ارتباط برقرار کند. و وزنی معادل عدد **یک** برای آنها در نظر گرفته شده است.

- **متوسط**: سیستم‌های جداگانه‌ای هستند که با سیستم جاری از طریق پروتکل‌های ارتباطی استاندارد مانند TCP/IP، FTP یا HTTP تعامل دارند، یا پایگاه‌داده خارجی که می‌توان به آن با استفاده از SQLاستاندارد دسترسی پیدا کرد. . و وزنی معادل عدد **دو** برای آنها در نظر گرفته شده است.

- **پیچیده**: معمولاً کاربران نهایی هستند که با سیستم ارتباط برقرار می‌کنند. . و وزنی معادل عدد سه برای آنها در نظر گرفته شده است.

UAW (Unadjusted Actor Weighting) : پس از دسته‌بندی actor ها ، مدیر پروژه تعداد آنها را در هر دسته شمارش کرده و مقادیر را در جدول وزن‌دهی actor ها بدون تنظیم وارد می‌کند. سپس مجموع وزن‌دهی actor های بدون تنظیم را محاسبه می‌کند. این مقدار با جمع کردن نتایج فردی به دست می‌آید. Unadjusted Actor Weight Total

برآورد زمان لازم برای توسعه یک نرم افزار

مثال:

فرض کنیم که برای Use case مد نظر ما، تعداد Actor ساده صفر نفر، Actor متوسط صفر نفر و Actor پیچیده چهار نفر در نظر گرفته شده است که با سیستم در حال توسعه تعامل خواهند داشت، UAW کل برابر با ۱۲ خواهد بود.

UAW=12

Unadjusted Actor Weighting Table:

Actor Type	Description	Weighting Factor	Number	Result
Simple	External system with well-defined API	1	0	0
Average	External system using a protocol-based interface, e.g., HTTP, TCT/IP, or a database	2	0	0
Complex	Human	3	4	12
			Unadjusted Actor Weight Total (UAW)	**12**

برآورد زمان لازم برای توسعه یک نرم افزار

Use Case :

یک use case معمولا نمایانگر یک فرایند کسب‌وکار است که سیستم انجام می‌دهد ،بسته به تعداد تراکنش‌های منحصر به فردی که یک use case باید به آنها پاسخ دهد، می‌توان آن را به سه نوع زیر تقسیم بندی کرد:

- **ساده** : یک use case در صورتی که از یک تا سه تراکنش پشتیبانی کند به عنوان ساده در نظر گرفته می شود و وزنی برابر ۵ به آن اختصاص داده می شود.

- **متوسط**: اگر از چهار تا هفت تراکنش پشتیبانی کند به عنوان متوسطدر نظر گرفته می شود و وزنی برابر ۱۰ به آن اختصاص داده می شود.

- **پیچیده**: اگر بیش از هفت تراکنش را پشتیبانی کند به عنوان پیچیده در نظر گرفته می شود و وزنی برابر ۱۵ به آن اختصاص داده می شود.

UUCW (Unadjusted Use-Case Weight) : پس از دسته‌بندی use caseها، مدیر پروژه مقدار کلی UUCW را با ضرب کردن تعداد هر نوع در وزن‌های مناسب و جمع کردن نتایج، بدست می آورد.

UUCW=70

Unadjusted Use-Case Weighting Table:

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1–3 transactions	5	3	15
Average	4–7 transactions	10	4	40
Complex	>7 transactions	15	1	15
			Unadjusted Use Case Weight Total (UUCW)	**70**

برآورد زمان لازم برای توسعه یک نرم افزار

سپس مدیر پروژه مقدار unadjusted use case point (UUCP) را با جمع کردن مجموع UAW و مجموع UUCW محاسبه می‌کند.

Unadjusted Use Case Points (U**UCP**) = UAW + UUCW

UUCP → 12+70=82

حال می توانیم Use case point های برا اساس عوامل پیچیدگی فنی (TCF) و عوامل محیطی (EF) **تنظیم** کنیم.

هدف این عوامل این است که پروژه برای پیچیدگی های فنی سیستم در حال توسعه و سطوح تجربه کارکنان، مورد ارزیابی قرار گیرند.

هر یک از این عوامل یک مقدار بین ۰ تا ۵ دارد، که ۰ نشان‌دهنده این است که عامل برای سیستم مورد نظر بی‌اهمیت است و ۵ نشان‌دهنده این است که عامل برای موفقیت سیستم ضروری است.

مقادیر اختصاص داده شده سپس در وزن‌های مربوطه ضرب می‌شوند و نتایج برای ایجاد یک مقدار عامل فنی یا TFactorو یک مقدار عامل محیطی یا EFactorجمع می‌شوند.

محاسبه عوامل پیچیدگی فنی Use-case point ها (TCF)

۱. توزیع‌پذیری سیستم
۲. اهمیت زمان پاسخگویی
۳. سطح کارایی کاربر نهایی
۴. پیچیدگی پردازش داخلی سیستم
۵. اهمیت استفاده مجدد از کد
۶. سهولت در فرآیند نصب
۷. سهولت استفاده از سیستم
۸. قابلیت انتقال به پلتفرم‌های دیگر
۹. اهمیت نگهداری سیستم
۱۰. نیاز به پردازش موازی و هم‌زمان
۱۱. سطح امنیت ویژه مورد نیاز
۲۱. سطح دسترسی سیستم توسط third partie ها
۳۱. نیاز به آموزش خاص برای کاربران نهایی

محاسبه عوامل پیچیدگی فنی Use-case point ها

با فرض مقادیر مشخص برای این عوامل، مقدار TFactorبه‌عنوان مجموع وزنی این عوامل محاسبه می‌شود. برای مثال، اگر مقادیر T1 تا T13به ترتیب :

T1 (0), T2 (5), T3 (3), T4 (1), T5 (1), T6 (2), T7 (4), T8 (0), T9 (2), T10 (0), T11 (0), T12 (0), T13 (0)

باشند، TFactor برابر ۱۵ خواهد بود. این مقدار سپس در معادله TCF (Technical Complexity Factor) قرار می‌گیرد تا TCF نهایی به‌دست آید.

Technical Complexity Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
T1	Distributed system	2.0	0	0	
T2	Response time or throughput performance objectives	1.0	5	5	
T3	End-user online efficiency	1.0	3	3	
T4	Complex internal processing	1.0	1	1	
T5	Reusability of code	1.0	1	1	
T6	Ease of installation	0.5	2	1	
T7	Ease of use	0.5	4	2	
T8	Portability	2.0	0	0	
T9	Ease of change	1.0	2	2	
T10	Concurrency	1.0	0	0	
T11	Special security objectives included	1.0	0	0	
T12	Direct access for third parties	1.0	0	0	
T13	Special user training required	1.0	0	0	
			Technical Factor Value (TFactor)	15	

Technical Complexity Factor (TCF) = 0.6 + (0.01 * TFactor) → 0.6 + (0.01 * 15)= 0.75 TCF=0.75

محاسبه عوامل محیطی Use-case point ها (EF)

۱. سطح تجربه تیم توسعه در فرایند مورد استفاده
۲. سطح تجربه توسعه اپلیکیشن
۳. سطح تجربه شی‌گرایی
۴. سطح قابلیت تحلیل‌گر اصلی
۵. سطح انگیزه تیم توسعه
۶. ثبات نیازها
۷. گنجاندن کارمندان پاره‌وقت در تیم توسعه
۸. دشواری زبان برنامه‌نویسی مورد استفاده

محاسبه عوامل محیطی Use-case point ها

با فرض مقادیر مشخص برای عوامل E1تا E8، مقدار EFactorبه‌دست می‌آید. اگر مقادیر فرضی ۴، ۴، ۴، ۵، ۵، ۵، ۰ و ۴ باشد:

EFactor برابر ۲۵,۵ خواهد بود. سپس این مقدار در معادله EF (Environmental Factor) قرار می‌گیرد تا EFنهایی به‌دست آید.

Environmental Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
E1	Familiarity with system development process being used	1.5	4	6	
E2	Application experience	0.5	4	2	
E3	Object-oriented experience	1.0	4	4	
E4	Lead analyst capability	0.5	5	2.5	
E5	Motivation	1.0	5	5	
E6	Requirements stability	2.0	5	10	
E7	Part-time staff	–1.0	0	0	
E8	Difficulty of programming language	–1.0	4	–4.0	
			Environmental Factor Value (EFactor)	**25.5**	

Environmental Factor (EF) = 1.4 + (-0.03 * EFactor) → 1.4 + (-0.03 * 25.5)=0.635 **EF=0.635**

محاسبه Use-case point

در نهایت، با استفاده از مقادیر TCFو EFو همچنین UUCP، می‌توان مقدار UCP را با استفاده از معادله زیر به‌دست آورد:

Adjusted Use Case Points (UCP) = UUCP * TCF * ECF → 70 * 0.75 * 0.635= 33.3375

UCP=33.3375

این فرآیند به مدیران پروژه کمک می‌کند تا تخمین دقیق‌تری از تلاش و زمان مورد نیاز برای تکمیل پروژه داشته باشند.

پس از محاسبه اندازه تخمینی UCP، حالا تلاش و زمان مورد نیاز برای ساخت سیستم را برآورد کنیم.

تخمین تلاش: در ابتدا، پیشنهاد می شد که UCP را در ۲۰ ضرب کنیم تا تخمین بزنیم که چند ساعت کاری برای توسعه سیستم لازم است.

ضریب ساعت‌های کاری (PHM) : با گذشت زمان، تجربه‌ها نشان دادند که می‌توان از دو مقدار (۲۰ یا ۲۸) برای PHM استفاده کرد. این تصمیم‌گیری بر اساس ارزش‌های اختصاص‌یافته به عوامل محیطی (Efactors) انجام می‌شود.

قاعده تصمیم‌گیری برای PHM:

اگر مجموع ارزش‌های عوامل E1تا E6(که باید کمتر از ۳ باشند) و عوامل E7و E8(که باید بزرگتر از ۳ باشند) کمتر از ۲ باشد، PHMبرابر ۲۰ است.

اگر مجموع برابر با ۳ یا ۴ باشد، PHMبرابر ۲۸ است.

در غیر این صورت، به دلیل خطر بالای شکست پروژه، باید دوباره آن را بررسی کرد.

If the sum of (number of Efactors E1 through E6 assigned value < 3) and
(number of Efactors E7 and E8 assigned value > 3)
$\leqslant 2$
PHM = 20
Else If the sum of (number of Efactors E1 through E6 assigned value < 3) and
(number of Efactors E7 and E8 assigned value > 3)
$= 3$ or 4
PHM = 28
Else
Rethink project; it has too high of a risk for failure

هیچ یک از عوامل E1تا E6 ارزش کمتری از ۳ نداشتند. (0)
تنها یک عامل E8 ارزشی بالاتر از ۳ داشت.(1)
بنابراین، مجموع دو ارزش فوق ۱ شد و در نتیجه، PHM برابر با **۲۰** انتخاب شد.

محاسبه تعداد ساعت‌های کاری: با استفاده از UCP که ۳۳,۳۳۷۵ بود و PHM که ۲۰ است، می‌توان تعداد ساعت‌های کاری تخمینی را محاسبه کرد:

Effort in person-hours = UCP * PHM 20 * 33.3375=**666.75**

نتیجه

بنابراین، در نهایت نتیجه می‌گیریم که برای ساخت سیستم، تخمین زده می‌شود (با توجه به برآورد منابع، مدیریت زمان،قیمت پروژه،پیچیدگی ها) که به **۶۶۶.۷۵ ساعت کاری** نیاز است. این اطلاعات به مدیران پروژه کمک می‌کند تا منابع و زمان لازم برای پروژه را بهتر برنامه‌ریزی کنند.

Unadjusted Actor Weighting Table:

Actor Type	Description	Weighting Factor	Number	Result
Simple	External system with well-defined API	1	0	0
Average	External system using a protocol-based interface, e.g., HTTP, TCT/IP, or a database	2	0	0
Complex	Human	3	4	12
			Unadjusted Actor Weight Total (UAW)	**12**

Unadjusted Use-Case Weighting Table:

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1–3 transactions	5	3	15
Average	4–7 transactions	10	4	40
Complex	>7 transactions	15	1	15
			Unadjusted Use Case Weight Total (UUCW)	**70**

Unadjusted Use-Case Points (UUCP) = UAW + UUCW **82 = 12 + 70**

Technical Complexity Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
T1	Distributed system	2.0	0	0	
T2	Response time or throughput performance objectives	1.0	5	5	
T3	End-user online efficiency	1.0	3	3	
T4	Complex internal processing	1.0	1	1	
T5	Reusability of code	1.0	1	1	
T6	Ease of installation	0.5	2	1	
T7	Ease of use	0.5	4	2	
T8	Portability	2.0	0	0	
T9	Ease of change	1.0	2	2	
T10	Concurrency	1.0	0	0	
T11	Special security objectives included	1.0	0	0	
T12	Direct access for third parties	1.0	0	0	
T13	Special user training required	1.0	0	0	
			Technical Factor Value (TFactor)	**15**	

Technical Complexity Factor (TCF) = 0.6 + (0.01 * TFactor) **0.75 = 0.6 + (0.01 * 15)**

Environmental Factors:

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
E1	Familiarity with system development process being used	1.5	4	6	
E2	Application experience	0.5	4	2	
E3	Object-oriented experience	1.0	4	4	
E4	Lead analyst capability	0.5	5	2.5	
E5	Motivation	1.0	5	5	
E6	Requirements stability	2.0	5	10	
E7	Part-time staff	–1.0	0	0	
E8	Difficulty of programming language	–1.0	4	–4.0	
			Environmental Factor Value (EFactor)	**25.5**	

Environmental Factor (EF) = 1.4 + (−0.03 * EFactor) **0.635 = 1.4 + (−0.03 * 25.5)**
Adjusted Use Case Points (UCP) = UUCP * TCF * ECF **33.3375 = 70 * 0.75 * 0.635**
Effort in person-hours = UCP * PHM **666.75 = 20 * 33.3375**

چند کاربرد محاسبه تعداد ساعات کار مورد نیاز طبق موارد اسلایدهای قبل:

۱- برنامه‌ریزی منابع

اگر تخمین زده شود که پروژه به ۶۶۶,۷۵ ساعت نیاز دارد و شما ۳ برنامه‌نویس دارید، می‌توانید محاسبه کنید:

تعداد ساعات کاری هر برنامه‌نویس: (666.75)/3=222.25

۲- تخمین هزینه‌ها

اگر نرخ دستمزد هر برنامه‌نویس ۱۵۰هزار تومن در ساعت باشد:

هزینه کل پروژه: (666.75)*150000=100000000

۳- شناسایی ریسک‌ها

اگر قبلا پیش‌بینی شده باشد که پروژه ۴۰۰ ساعت نیاز دارد ولی محاسبات نشان می‌دهد که ۶۶۶,۷۵ ساعت نیاز است، این افزایش می‌تواند نشانه‌ای از ریسک بالا و نیاز به بازنگری در برنامه‌ریزی باشد.

۴- مقایسه با پروژه‌های مشابه

فرض کنید پروژه‌های مشابه در گذشته به طور میانگین ۵۰۰ ساعت نیاز داشتند. اگر این پروژه ۶۶۶,۷۵ ساعت برآورد شود، می‌توانید نتیجه بگیرید که این پروژه پیچیده‌تر از پروژه‌های قبلی است.

۵- بهبود فرآیندها

اگر در پروژه‌های قبلی ۶۰۰ ساعت تخمین زده شده و ۷۰۰ ساعت واقعی مصرف شده، می‌توانید فرآیندها را بررسی کنید تا ببینند کجا زمان بیشتری صرف شده است.

۶- پیش‌بینی زمان تحویل

با ۶۶۶٫۷۵ ساعت و فرض اینکه ۳ برنامه‌نویس به طور همزمان کار می‌کنند، می‌توانید زمان تقریبی تحویل را پیش‌بینی کنید:

زمان تحویل: (666.75)/(3*40)=5.56 (هفته) (۴۰ ساعت کار در هفته)

۷- ارزیابی کارایی تیم

اگر پس از اتمام پروژه، ۸۰۰ ساعت واقعی مصرف شده باشد، می‌توانید کارایی تیم را ارزیابی کنید:

کارایی: (666.75/800)*100=83.34%

ایجاد و مدیریت برنامه کاری

پس از عملیات فوق حال نوبت به آن رسیده است که مدیر پروژه یک برنامه کاری **(Workplan)** ایجاد می‌کند.

Workplan: که یک جدول زمانی پویا است و تمام وظایفی را که باید در طول پروژه انجام شوند، ثبت و پیگیری می‌کند.

این برنامه کاری شامل هر وظیفه و اطلاعات مهمی درباره آن، از جمله زمان اتمام آن، فرد مسئول انجام کار و هر محصولی که به دست می‌آید، است.

سطح جزئیات و مقدار اطلاعات موجود در **Workplan** بستگی به نیازهای پروژه دارد و معمولاً با پیشرفت پروژه، جزئیات بیشتر می‌شوند.

وظیفه مدیر پروژه : اهداف کلی سیستم باید در نیازمندی های سیستم ذکر شوند و وظیفه مدیر پروژه، شناسایی تمام وظایفی است که باید برای رسیدن به آن اهداف انجام شوند.

چگونه می‌توان فهمید که چه کارهایی باید انجام شود تا سیستمی که قبلاً ساخته نشده، ساخته شود؟

ایجاد و مدیریت برنامه کاری(workplan)

روش‌ها برای شناسایی وظایف:

- دریافت و اصلاح فهرستی از وظایفی است که قبلاً توسعه یافته.
- **فهرست‌های استاندارد** وظایف یا متدولوژی‌هایی وجود دارند که می‌توانند به عنوان نقطه شروع مورد استفاده قرار گیرند.

چون از یک متدولوژی مبتنی بر فرآیند یکپارچه استفاده می‌کنیم، می‌توانیم از فازها، جریان‌های کاری(workflows) و تکرارها (iterations)به عنوان نقطه شروع برای ایجاد یک ساختار شکست کار تکاملی (WBS)و یک برنامه کاری تکراری(Iterative Workplan) استفاده کنیم.

ایجاد و مدیریت برنامه کاری(workplan)

ساختارهای شکسته کار تکاملی و برنامه‌های کاری تکراری

برای برنامه‌ریزی پروژه‌های مبتنی بر شیء گرائی می توان از روش های مبتنی بر توسعه تدریجی و تکراری استفاده کرد.

ویژگی‌های WBS تکاملی:

- **سازماندهی استاندارد**:

ساختار شکسته کار تکاملی استاندارد (WBS) به این صورت پیاده سازی می شوند که ابتدا جریان‌های کار(**workflows**)، سپس مراحل(**phases**) و بعد وظایف خاصی(**specific tasks**) که در هر تکرار انجام می‌شوند، مشخص می‌شوند.

- **توسعه تدریجی و تکراری**:

WBSبه گونه‌ای طراحی می‌شود که به تدریج و به صورت تکراری به‌روزرسانی شود. این کار به مدیران پروژه کمک می‌کند که هزینه‌ها و زمان را به‌طور واقع‌بینانه‌تری برآورد کنند.

- **مقایسه پذیری**:

ساختار WBSبه هیچ پروژه خاصی وابسته نیست، بنابراین می‌توان پروژه جدید را با پروژه‌های قبلی مقایسه کرد. این به تحلیل‌گران کمک می‌کند که از موفقیت‌ها و شکست‌های گذشته یاد بگیرند.

ایجاد و مدیریت برنامه کاری(workplan)

نحوه ایجاد :WBS

- جریان‌های کار(Workflows):

ابتدا باید فعالیت‌ها یا مراحل کلیدی پروژه که نمایانگر بخش‌های اصلی آن هستند، شناسایی و تعریف شوند.

مثلا: برنامه ریزی، تحلیل نیازمندی‌ها، طراحی، پیاده سازی

- تجزیه به مراحل(Phases):

هر یک از جریان‌های کاری که در پروژه شناسایی شده‌اند، به بخش‌های کوچک‌تر و قابل مدیریت‌تری تقسیم می‌شوند که به عنوان مراحل مختلف فرآیند پروژه شناخته می‌شوند.

مثال، برای جریان کار "تحلیل نیازمندی‌ها"، مراحل می‌تواند شامل مواردی مانند:

جمع‌آوری نیازها: شامل برگزاری جلسات با ذینفعان و جمع‌آوری اطلاعات.

تحلیل نیازها: بررسی و اولویت‌بندی نیازها.

مستندسازی نیازها: نوشتن و تایید مستندات مربوط به نیازها.

- تجزیه به وظایف(Specific tasks):

وظایف، کارهایی هستند که باید برای رسیدن به اهداف هر مرحله انجام شوند. مثلا برگزاری جلسات با ذینفعان، تهیه پرسشنامه

در نهایت، با تکمیل هر تکرار در فرآیند توسعه، وظایف و تکرارهای جدید به WBSاضافه می‌شوند. این بدین معنی است که WBSبه طور مداوم با اطلاعات جدید به‌روزرسانی می‌شود و در نتیجه، انعطاف‌پذیری و کارایی پروژه افزایش می‌یابد.

مثال:

I. مدل‌سازی کسب‌وکار

الف. آغاز:

۱. درک وضعیت فعلی کسب‌وکار (۰,۵ روز)
۲. کشف مشکلات فرآیندهای کسب‌وکار (۰,۲۵ روز)
۳. شناسایی پروژه‌های احتمالی (۰,۲۵ روز)

ب. بسط(توسعه)

ج. ساخت

د. انتقال

ه. تولید

II. نیازمندی‌ها

الف. آغاز:

۱. شناسایی تکنیک مناسب تحلیل نیازمندی‌ها (۰,۲۵ روز)
۲. شناسایی تکنیک‌های مناسب جمع‌آوری نیازمندی‌ها (۰,۲۵ روز)
۳. شناسایی نیازمندی‌های کاربردی و غیرکاربردی
    الف. برگزاری جلسات JAD (3 روز)
    ب. تحلیل اسناد (۵ روز)
    ج. برگزاری مصاحبه‌ها
        مصاحبه با حامی پروژه (۰,۵ روز)
        مصاحبه با تماس سیستم انبار (۰,۵ روز)
        مصاحبه با تماس سیستم سفارش خاص (۰,۵ روز)
        مصاحبه با تماس ISP (0.5 روز)
        مصاحبه با تماس وب انتخاب CD (0.5 روز)
        مصاحبه با دیگر کارکنان (۱ روز)
    د. مشاهده فرآیندهای فروشگاه خرده‌فروشی (۰,۵ روز)
۴. تحلیل سیستم‌های فعلی (۴ روز)
۵. ایجاد تعریف نیازمندی‌ها
    الف. تعیین نیازمندی‌هایی که باید پیگیری شوند (۱ روز)
    ب. جمع‌آوری نیازمندی‌ها به‌محض دریافت آن‌ها (۵ روز)
    ج. بازبینی نیازمندی‌ها با حامی (۲ روز)

ب. بسط

ج. ساخت

د. انتقال

ه. تولید

	Duration	Dependency
I. Business Modeling		
a. Inception		
1. Understand current business situation	0.50 days	
2. Uncover business process problems	0.25 days	
3. Identify potential projects	0.25 days	
b. Elaboration		
c. Construction		
d. Transition		
e. Production		
II. Requirements		
a. Inception		
1. Identify appropriate requirements-analysis technique	0.25 days	
2. Identify appropriate requirements-gathering techniques	0.25 days	
3. Identify functional and nonfunctional requirements		II.a.1, II.a.2
A. Perform JAD sessions	3 days	
B. Perform document analysis	5 days	II.a.3.A
C. Conduct interviews		II.a.3.A
1. Interview project sponsor	0.5 days	
2. Interview inventory system contact	0.5 days	
3. Interview special order system contact	0.5 days	
4. Interview ISP contact	0.5 days	
5. Interview CD Selection Web contact	0.5 days	
6. Interview other personnel	1 day	
D. Observe retail store processes	0.5 days	II.a.3.A
4. Analyze current systems	4 days	II.a.1, II.a.2
5. Create requirements definition		II.a.3, II.a.4
A. Determine requirements to track	1 day	
B. Compile requirements as they are elicited	5 days	II.a.5.A
C. Review requirements with sponsor	2 days	II.a.5.B
b. Elaboration		
c. Construction		
d. Transition		
e. Production		

Managing Scope

دامنه پروژه: مشخص می‌کند که چه چیزی در پروژه گنجانده شده و چه چیزی خارج از آن است.

خزش دامنه:

به معنای اضافه شدن نیازهای جدید پس از تعریف دامنه اصلی است که می‌تواند به دلایل مختلفی مانند درک جدید کاربران یا تصمیمات مدیریتی رخ دهد. این امر می‌تواند منجر به افزایش هزینه و زمان پروژه شود.

گاهی اوقات تغییرات نمی‌توانند به سیستم موجود اضافه شوند حتی اگر واقعاً مفید باشند. در این صورت، این افزودنی‌ها باید به‌عنوان بهبودهای آینده برای سیستم ثبت شوند. مدیر پروژه می‌تواند پیشنهاد کند که قابلیت‌هایی در نسخه‌های آینده سیستم ارائه شود و بدین ترتیب از گفتن "نه" به کسی اجتناب کند.

برای مقابله با این مشکل، استفاده از فرآیندهای توسعه تکراری و افزایشی موثر است. در ابتدای پروژه باید نیازها به دقت شناسایی شوند و اگر نیازهای جدید ضروری باشند، باید عواقب آن‌ها ارزیابی و مستند شود.

تکنیک‌های چابک مانند جلسات روزانه اسکرام(Scrum) و لیست موجودی محصول(Product Backlog) نیز به مدیریت دامنه کمک می‌کنند. جلسات روزانه به تیم امکان می‌دهند تا از وضعیت پروژه مطلع شوند و لیست موجودی محصول به عنوان فهرستی اولویت‌بندی‌شده از نیازها عمل می‌کند. به این ترتیب، تیم همیشه بر روی نیازهای بحرانی تمرکز دارد و مدیریت تغییرات ساده‌تر می‌شود.

Timeboxing

زمان‌بندی یک روش برای مدیریت پروژه‌هاست که بر اساس آن یک مهلت ثابت برای تحویل پروژه تعیین می‌شود. این روش به جای تمرکز بر روی تعداد وظایف، به زمان اهمیت می‌دهد.

نکات کلیدی:

مهلت ثابت: یک تاریخ تحویل مشخص تعیین می‌شود و یک قسمت از کار باید تا آن زمان تکمیل شود.

تمرکز بر اولویت‌ها: تیم پروژه باید ویژگی‌های مهم را شناسایی و اولویت‌بندی کند تا مهم‌ترین نیازها در مهلت مشخص شده تأمین شود.

کیفیت: کیفیت محصول نباید تحت تأثیر زمان قرار گیرد. زمان آزمایش و بررسی باید حفظ شود.

تکرار و بهبود: ممکن است نیاز به دوره‌های آینده برای اضافه کردن ویژگی‌های بیشتر وجود داشته باشد.

Timeboxing

مثال :

- یک نرم‌افزار واژه‌پرداز را در نظر بگیرید. در ۸۰ درصد از زمان، تنها ۲۰ درصد از ویژگی‌ها، مانند بررسی املا، برجسته‌سازی و برش و چسباندن، استفاده می‌شوند. سایر ویژگی‌ها، مانند ادغام سند و ایجاد برچسب‌های پستی، ممکن است جذاب باشند، اما جزو نیازهای روزمره نیستند. همین امر درباره دیگر برنامه‌های نرم‌افزاری صدق می‌کند؛ بیشتر کاربران فقط به مجموعه‌ای کوچک از قابلیت‌های آن‌ها وابسته‌اند.

- اسپرینت در اسکرام (یک دوره زمانی دو هفته ای)

Refining Estimates

اصلاح تخمین‌ها

تخمین‌هایی که در ابتدای پروژه انجام می‌شوند، باید به مرور زمان اصلاح شوند. این بدان معنا نیست که تخمین‌ها در ابتدا اشتباه بوده‌اند، بلکه در شروع کار، اطلاعات کافی برای یک ارزیابی دقیق وجود ندارد.

تخمین‌های ابتدایی: در مرحله برنامه‌ریزی، مدیر پروژه و حامی آن سعی می‌کنند بگویند پروژه چقدر زمان و هزینه خواهد داشت. اما در این مرحله، اطلاعات کمی درباره سیستم وجود دارد.

دقت بیشتر با پیشرفت پروژه: هرچه پروژه پیش می‌رود و اطلاعات بیشتری جمع‌آوری می‌شود، تخمین‌ها دقیق‌تر می‌شوند.

حاشیه خطا: یک پروژه می‌تواند ۱۰۰ درصد **حاشیه خطا** برای هزینه و ۲۵ درصد برای زمان داشته باشد. یعنی اگر هزینه یک پروژه صد میلیون تومن پیش‌بینی شود و بیست هفته کاری، در واقع ممکن است دویست میلیون تومن هزینه و بیست و پنج هفته زمان نیاز داشته باشد.

تنظیم تخمین‌ها: اگر بخشی از پروژه بیش از زمان پیش‌بینی شده طول بکشد، باید تخمین‌های آینده را بر اساس این تاخیر تغییر دهند. مثلاً اگر فاز اول ۱۰ درصد بیشتر از زمان تعیین شده طول بکشد، باید سایر تخمین‌ها نیز ۱۰ درصد افزایش یابد.

در نهایت، اصلاح تخمین‌ها به تیم کمک می‌کند تا واقع‌گرایانه‌تر به برنامه‌ریزی و اجرای پروژه ادامه دهد.

Managing Risk

مدیریت ریسک : شناسایی و رسیدگی به خطرات مرتبط با یک پروژه. این خطرات می‌توانند ناشی از عوامل مختلفی مانند کارکنان ضعیف، افزایش ناخواسته محدوده کار، طراحی نادرست و تخمین‌های خوش‌بینانه باشند.

ارزیابی ریسک: تیم‌های پروژه معمولاً یک سند به نام ارزیابی ریسک تهیه می‌کنند که شامل ریسک‌های احتمالی، احتمال وقوع آنها و تأثیرشان بر پروژه است.

راه‌های مقابله: برای هر ریسک، راه‌های مقابله وجود دارد، مانند آموزش اعضای تیم برای از بین بردن کمبود مهارت.

اولویت‌بندی ریسک‌ها: مدیران پروژه باید ریسک‌ها را بر اساس اندازه و اهمیت آنها اولویت‌بندی کنند و به‌روزرسانی‌های منظم انجام دهند.

هدف نهایی جلوگیری از تأثیر ریسک‌ها بر زمان و هزینه پروژه است.

Managing Risk

مثال:

ارزیابی ریسک

ریسک ۱:

توسعه این سیستم احتمالا به میزان قابل توجهی کند خواهد شد زیرا اعضای تیم پروژه، قبل از این پروژه، در جاوا برنامه نویسی نکرده اند.

احتمال خطر:

احتمال ریسک بالا

تاثیر بالقوه این ریسک بر پروژه:

این ریسک احتمالاً زمان تکمیل وظایف برنامه نویسی را تا ۵۰ درصد افزایش می دهد.

راه های مقابله با این خطر:

زمان و منابعی به آموزش مقدماتی جاوا برای برنامه نویسانی که برای این پروژه استفاده می شوند اختصاص داده شود.

افراد جدید جاوا کار به تیم اضافه شوند و افراد قبلی که سابقه کار در این تیم را داشته اند ولی جاوا کار نکرده اند به عنوان راهنما در کنار افراد جدید باشند.

ریسک ۲: ....

کارکنان پروژه

یکی از کارهای مدیر پروژه تعیین تعداد افرادی است که باید به پروژه اختصاص داده شوند، همچنین تطبیق مهارت‌های افراد با نیازهای پروژه، انگیزه‌بخشی به آن‌ها برای رسیدن به اهداف پروژه و کاهش درگیری‌هایی است که در طول زمان ممکن است پیش آید.

تیم ژله ای (jelled team) یعنی گروهی از افراد که بسیار هماهنگ و هم‌نوا هستند و نتیجه کارشان بیشتر از مجموع تلاش‌های فردی‌شان است.

کارکنان پروژه

ویژگی‌های تیم ژله ای:

– **پایداری اعضا**: تغییر اعضا در طول پروژه بسیار کم است. اعضای تیم احساس مسئولیت عمیقی نسبت به یکدیگر دارند و ترک تیم را به معنای شکست این مسئولیت می‌دانند.

– **احساس هویت قوی**: این تیم‌ها یک هویت خاص دارند. اعضا نه تنها یک نام برای گروه انتخاب می‌کنند، بلکه به نوعی در کنار هم به یک موجودیت منحصربه‌فرد تبدیل می‌شوند. آن‌ها ممکن است فعالیت‌های غیرکاری مثل ناهار خوردن یا ورزش کردن با هم انجام دهند.

– **احساس نخبگی**: اعضای تیم جلب شده احساس می‌کنند که در مقایسه با سایر افراد در محیط کار، در موقعیت برتری هستند. این حس را می‌توان در تیم‌های ورزشی یا تیم‌های ویژه نظامی مشاهده کرد، جایی که هر عضو در تخصص خود ماهر است و به توانایی‌های دیگر اعضا اعتماد دارد.

– **احساس مالکیت بر پروژه**: اعضای تیم احساس می‌کنند که سیستم اطلاعاتی در حال توسعه متعلق به تیم است و نه به یک فرد خاص. در واقع، خروجی تیم مهم‌تر از سهم فردی هر عضو است.

– **لذت از کار**: اعضای تیم جلب شده واقعاً از کار کردن با هم لذت می‌برند و این به چالش‌های پروژه مرتبط است. اگر پروژه چالش‌برانگیز باشد، اعضا بیشتر از آن لذت خواهند برد.

کارکنان پروژه

پرهیز از ناهنجاری‌های تیم

زمانی که یک تیم ژله ای می‌شود، از پنج ناهنجاری تیمی پرهیز می‌کند:

عدم اعتماد: عدم اعتماد باعث ناهنجاری تیم می‌شود. در تیم‌های ژله ای، اعضا از مطرح کردن مشکلات و درخواست کمک نمی‌ترسند و این موجب افزایش اعتماد بین آن‌ها می‌شود.

عدم تعهد: در تیم‌های ناهنجار، اعضا بیشتر به عملکرد فردی خود توجه دارند. اما در تیم‌های جلب شده، تمرکز بر عملکرد کلی تیم است.

اجتناب از پاسخ‌گویی: در تیم‌های ناهنجار، اعضا از پاسخ‌گویی به یکدیگر فراری هستند. در تیم‌های ژله ای، هر عضو به طور طبیعی احساس مسئولیت و پاسخ‌گویی دارد.

عدم توجه به نتایج تیم: اعضای تیم‌های ناهنجار به نتایج تیم توجه نمی‌کنند و فقط بر اهداف فردی تمرکز دارند. در تیم‌های ژله ای، هدف‌ها هم‌راستا می‌شوند و تیم به نتایج دست می‌یابد.

کارکنان پروژه

تعیین تعداد متوسط کارکنان مورد نیاز برای پروژه یکی از گام های اولیه است که می بایست به درستی ودقت لازم انجام شود. اضافه کردن نیروی انسانی بیشتر برای کاهش زمان پروژه تصمیم درستی نیست، زیرا افزایش اعضا، هماهنگی و مدیریت را دشوارتر می‌کند. به همین دلیل، باید اندازه تیم را زیر **هشت تا ده نفر** نگه داشت و در صورت نیاز به افراد بیشتر، زیرگروه‌ها را ایجاد خواهیم کرد. این کار به مدیر پروژه کمک می‌کند تا ارتباطات مؤثرتری در تیم‌های کوچک ایجاد کند.

پس از تعیین تعداد افراد مورد نیاز برای پروژه، مدیر پروژه یک طرحی را ارئه می دهد که در آن نقش‌ها و ساختار پروژه را مشخص می‌کند.

معمولاً یک مدیر پروژه بر پیشرفت کلی نظارت می‌کند و هسته تیم شامل تحلیلگران است. برای مدیریت گروهی از تحلیلگران، یک رهبر عملیاتی و برای نظارت بر برنامه‌نویسان، یک رهبر فنی منصوب می‌شود.

ساختارهای مختلفی برای تیم‌های پروژه وجود دارد. پس از تعریف نقش‌ها، مدیر پروژه باید فکر کند که کدام افراد می‌توانند چه نقش را ایفا کنند. اغلب یک نفر چندین نقش را بر عهده دارد. در انتصابات، توجه به مهارت‌های فنی و بین‌فردی ضروری است.

مهارت‌های فنی در انجام کارهای فنی و درک نقش‌های تکنیکی مفید است، در حالی که مهارت‌های بین‌فردی شامل توانایی‌های ارتباطی با کاربران تجاری و مدیران ارشد می‌باشد.

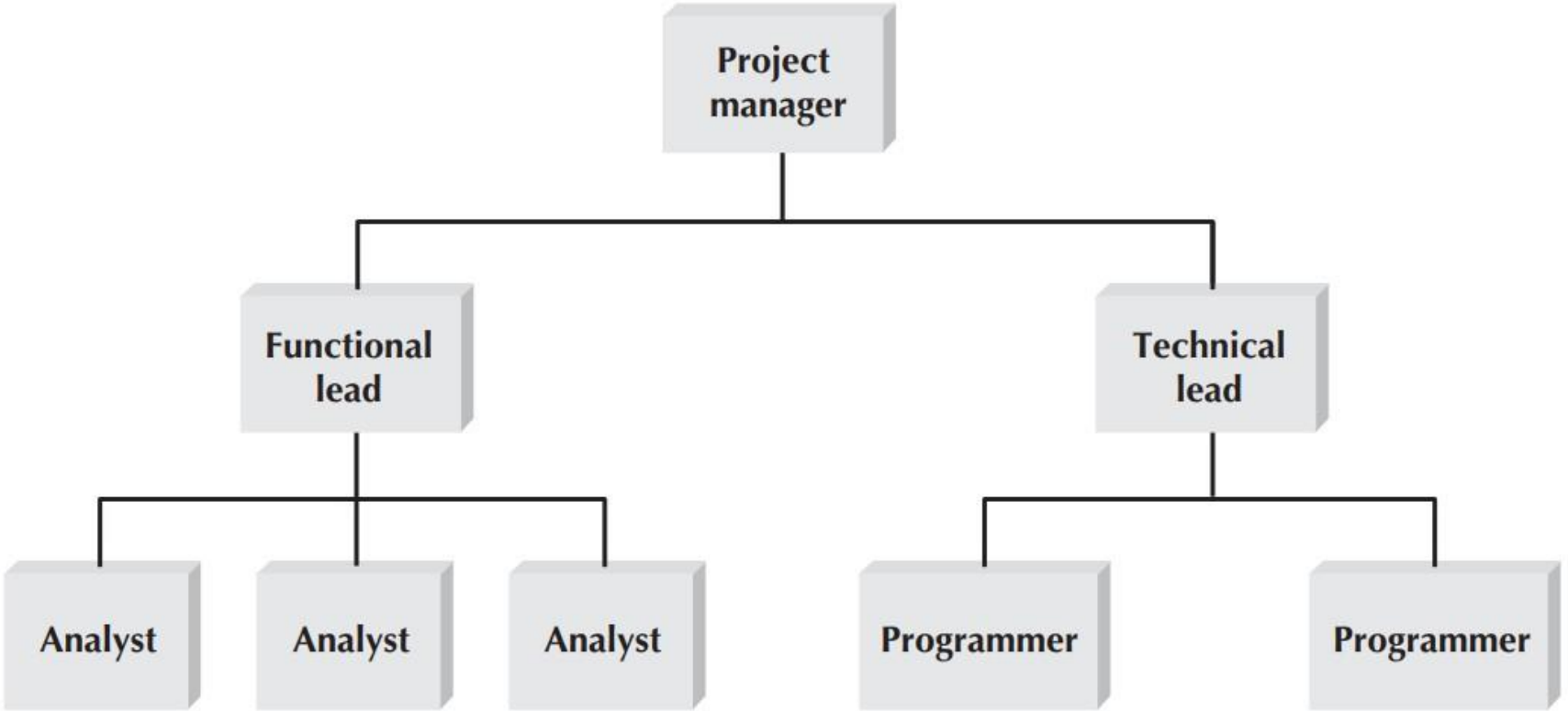
کارکنان پروژه

ایده‌آل این است که نقش‌ها با افرادی که مهارت‌های مناسب دارند پر شوند، اما ممکن است این افراد در دسترس نباشند یا در پروژه‌های دیگر مشغول باشند. بنابراین، انتساب اعضای تیم ترکیبی از یافتن افراد با مهارت‌های مناسب و در دسترس بودن آن‌ها است.

اگر مهارت‌های اعضای موجود با نیازهای پروژه مطابقت نداشته باشد، مدیر پروژه چند گزینه دارد:

- می‌توان افراد را از پروژه‌های دیگر خارج کرد که ممکن است اختلالات زیادی ایجاد کند.
- استفاده از کمک‌های خارجی مانند مشاوران برای آموزش اعضای تیم و شروع کار آن‌ها.
- مربی گیری (mentoring)می‌تواند گزینه‌ای دیگر باشد، یعنی یک عضو به پروژه مشابه دیگری فرستاده می‌شود تا مهارت‌های لازم را کسب کند و به تیم بازگردد.

کارکنان پروژه



Motivation

انگیزش

فقط اختصاص دادن افراد به وظایف کافی نیست؛ مدیران پروژه باید افراد را انگیزه دهند تا موفقیت پروژه تضمین شود. انگیزش به عنوان مهم‌ترین عامل تأثیرگذار بر عملکرد افراد شناخته شده است، اما تعیین چگونگی انگیزه‌دهی به تیم می‌تواند دشوار باشد. برخی مدیران فکر می‌کنند که پاداش‌های مالی بهترین راه برای انگیزه‌دهی هستند، اما بسیاری از مدیران معتقدند که این آخرین گزینه باید باشد. پاداش‌های مالی ممکن است انتظار بیشتری ایجاد کنند و اغلب به تنهایی مؤثر نیستند.

یکی از پیشنهادات **پینک** استفاده از "قاعده ۲۰ درصد" است، که به کارکنان اجازه می‌دهد ۲۰ درصد از زمان خود را به ایده‌هایی که به آن‌ها اعتقاد دارند اختصاص دهند. این پروژه‌ها نیازی به ارتباط با کار فعلی ندارند. برای مثال، Gmail و Google News. اگر ۲۰ درصد زیاد به نظر می‌رسد، پینک پیشنهاد می‌کند که با ۱۰ درصد شروع کنید.

پینک همچنین به جایزه‌های کوچک (Now That) اشاره می‌کند که کارکنان به همکاران خود به عنوان نشانه‌ای از قدردانی می‌دهند. این جوایز معمولاً مبلغ کمی دارند، اما ارزش آن‌ها در این است که از سوی همکاران اعطا می‌شوند.

همچنین باید بررسی شود که آیا کارکنان احساس تعلق به سازمان دارند یا خیر. اگر کارکنان به سازمان به عنوان "آن‌ها" اشاره کنند، ممکن است احساس بی‌تفاوتی داشته باشند، در حالی که اگر به عنوان "ما" اشاره کنند، نشان‌دهنده احساس تعلق به سازمان است.

پینک پیشنهاد می‌دهد که مدیران هر از چندگاهی یک روز را به خود کارکنان اختصاص بدهند تا بر روی هر چیزی که می‌خواهند کار کنند. این می‌تواند منجر به تولید نرم‌افزار جدید یا بهبود فرآیندهای داخلی شود. هدف این است که به اعضای تیم اجازه دهیم بر روی مشکلات جالب و چالش‌برانگیز تمرکز کنند.

او همچنین توصیه می‌کند که مسائل مربوط به حقوق از معادله انگیزش حذف شوند. به این معنا که همه کارکنان باید به اندازه کافی حقوق دریافت کنند تا پاداش‌های مالی موضوعیتی نداشته باشند. کارکنان فنی بیشتر از مسئولیت و فرصت یادگیری مهارت‌های جدید انگیزه می‌گیرند. پاداش‌های مالی ساده که به عنوان ناعادلانه تلقی شوند، می‌توانند انگیزش کلی تیم را کاهش دهند.

Motivation

یکی دیگر از پیشنهادات پینک این است که رهبر تیم از زبان کنترل‌کننده استفاده نکند، بلکه از اعضای تیم بخواهد که "به چیزی فکر کنند". به عبارت دیگر، رهبر تیم نباید اعتبار ایده‌ها را برای خود بگیرد، بلکه باید اعضای تیم را تشویق کند تا ایده‌ها را بررسی کنند.

پینک شواهدی ارائه می‌دهد که انگیزش درونی برای کارکنان دانش در قرن بیست و یکم بسیار مهم است. او پیشنهاد می‌کند که برای انگیزش افراد، باید به آن‌ها درجه‌ای از خودمختاری داد، از آن‌ها حمایت کرد تا بر حوزه تخصصی خود تسلط یابند، و آن‌ها را تشویق کرد تا پروژه‌هایی با هدف دنبال کنند.

حمایت از اعضای تیم برای تسلط بر حوزه تخصصی می‌تواند شامل تأمین مالی برای شرکت در کنفرانس‌ها و دوره‌های آموزشی باشد. همچنین، فراهم کردن محیط توسعه با کیفیت بالا، مانند نرم‌افزار و سخت‌افزار خاص برای توسعه برنامه‌های واقعیت مجازی، می‌تواند به تسلط بر تکنولوژی کمک کند.

نهایتاً، بسیار مهم است که اعضای تیم احساس کنند کار آن‌ها تأثیرگذار است. رهبر تیم باید آن‌ها را تشویق کند تا به مسائل مهمی که می‌تواند بر زندگی مردم تأثیر بگذارد، رسیدگی کنند.

Handling Conflict

مدیریت تضاد

مؤلفه بعدی، سازمان‌دهی پروژه به‌گونه‌ای است که تضاد میان اعضای گروه به حداقل برسد. انسجام گروهی، یعنی جاذبه‌ای که اعضا به گروه و یکدیگر احساس می‌کنند، تأثیر بیشتری بر بهره‌وری دارد تا توانایی‌ها یا تجربیات فردی اعضای پروژه.

تعریف واضح نقش‌ها در پروژه و مسئولیت‌پذیری اعضای تیم برای وظایفشان می‌تواند به کاهش تضادهای بالقوه کمک کند. برخی مدیران پروژه یک منشور پروژه ایجاد می‌کنند که هنجارها و قوانین پایه پروژه را فهرست می‌کند. به عنوان مثال، این منشور می‌تواند زمان‌های کار تیم، زمان برگزاری جلسات، نحوه ارتباط بین اعضا و رویه‌های به‌روزرسانی برنامه کاری را توضیح دهد.

استفاده از این منشور و تکنیک‌های دیگر در ابتدای پروژه می‌تواند به حفظ حداقل تضاد کمک کند.

ENVIRONMENT AND INFRASTRUCTURE MANAGEMENT

مدیریت محیط و زیرساخت

مدیریت محیط و زیرساخت به تیم توسعه در طول فرایند توسعه کمک می‌کند. این شامل انتخاب ابزارهای مناسب برای کار و تعیین استانداردهایی است که باید رعایت شوند.

همچنین، مدیریت زیرساخت به تعیین نوع و سطح مستندسازی که در طول پروژه باید انجام شود، می‌پردازد. این شامل فعالیت‌هایی مانند توسعه، تغییر و استفاده مجدد از کامپوننت های آماده، چارچوب‌ها و کتابخانه‌ها نیز می‌شود.

CASE Tools

ابزارهای CASE

ابزارهای مهندسی نرم‌افزار مبتنی بر رایانه (Computer-aided software engineering) فرایند توسعه نرم‌افزار را خودکار می‌کنند و شامل انواعی هستند که به مراحل تحلیل و طراحی کمک می‌کنند. این ابزارها مزایایی همچون افزایش سرعت انجام وظایف، متمرکز کردن مستندات، و ارائه اطلاعات به صورت نمودارهای قابل فهم دارند. همچنین می‌توانند هزینه‌های نگهداری را کاهش دهند و کیفیت نرم‌افزار را بهبود بخشند.

با این حال، این ابزارها نیاز به آموزش و تجربه دارند و باید توسط توسعه‌دهندگان آموزش‌دیده استفاده شوند تا مؤثر واقع شوند.

Standards

یکی از راه‌ها برای اطمینان از اینکه همه وظایف را به یک شکل انجام می‌دهند، **ایجاد** استانداردهایی است که تیم پروژه باید از آن‌ها پیروی کند. این استانداردها می‌توانند شامل قواعد رسمی برای نام‌گذاری فایل‌ها، فرم‌های مورد نیاز برای تکمیل اهداف، و راهنمایی‌های برنامه‌نویسی باشند.

استانداردها بهتر است در ابتدای هر مرحله مهم پروژه ایجاد و به وضوح به تمام اعضای تیم منتقل شوند. در طول پروژه، استانداردهای جدید در صورت لزوم اضافه می‌شوند. برخی از استانداردها در طول کل فرایند توسعه اعمال می‌شوند، در حالی که دیگران فقط برای وظایف خاص مناسب هستند. با پیروی از استانداردها، هماهنگی وظایف ساده‌تر شده و پروژه سریع‌تر تکمیل می‌شود.

استانداردها	
انواع استانداردها	نمونه
استانداردهای مستندسازی	تاریخ و نام پروژه باید به عنوان یک سربرگ در تمام اسناد ظاهر شود. تمام حاشیه ها باید روی ۱ اینچ تنظیم شوند. تمام موارد تحویلی باید به کلاسور پروژه اضافه شده و در فهرست مطالب آن ثبت شود.
استانداردهای کدنویسی	همه ماژول‌های کد باید شامل یک هدر باشد که برنامه‌نویس، آخرین تاریخ به‌روزرسانی و توضیح کوتاهی از هدف کد را فهرست کند. تورفتگی باید برای نشان دادن حلقه ها، عبارات if-then-else و عبارات case استفاده شود. به طور متوسط، هر برنامه باید شامل یک خط توضیح برای هر پنج خط کد باشد. هر دوشنبه پیشرفت واقعی کار را در برنامه کاری ثبت کنید صبح تا ساعت ۱۰ صبح
استانداردهای رویه ای	گزارش به جلسه به روز رسانی پروژه در روزهای شنبه ساعت ۱۵:۳۰. تمام تغییرات در یک سند الزامات باید توسط مدیر پروژه تایید شود.
استانداردهای طراحی رابط کاربری	لیبل ها به صورت متن پررنگ، در سمت چپ و به دنبال آن یک دونقطه ظاهر می شوند. ترتیب برگه های صفحه از بالا سمت چپ به سمت راست پایین حرکت می کند.

Documentation

مستندسازی

فرایند یکپارچه شامل مجموعه‌ای از مستندات استاندارد است که مورد انتظار است. این مستندات شامل درخواست سیستم، تحلیل امکان‌سنجی، نسخه‌های اصلی و بعدی برآورد تلاش، برنامه کاری در حال تحول و نمودارهای UMLبرای مدل‌های عملکردی، ساختاری و رفتاری است.

یک روش ضعیف در مدیریت پروژه، منتظر ماندن تا آخرین لحظه برای ایجاد مستندات است که معمولاً به مستنداتی منجر می‌شود که هیچ‌کس آن را درک نمی‌کند. تیم‌های خوب یاد می‌گیرند که تاریخچه سیستم را در حین تکامل آن مستند کنند، در حالی که جزئیات هنوز در خاطرشان هستند. بسیاری از ابزارهای CASE که از توسعه سیستم‌های شی‌گرا پشتیبانی می‌کنند، امکان اتوماسیون برخی از مستندات را فراهم می‌آورند.

اگرچه اکثر توسعه‌دهندگان از ایجاد مستندات متنفرند و این کار زمان‌بر است، اما این یک سرمایه‌گذاری خوب است که در بلندمدت بازدهی خواهد داشت.