**مدیریت پروژه چابک:**

**بهترین شیوه ها و روشها:**

**هنر مدیریت پروژه:**

همانطور که توسط گارتنر تعریف شده است ، مدیریت پروژه "استفاده از دانش ، مهارت ها ، ابزارها و تکنیک ها برای فعالیت های پروژه برای تأمین نیازهای پروژه" است.

مدیریت پروژه به عنوان بخشی جدایی ناپذیر از فرایندهای مهندسی نرم افزار همراه با تجزیه و تحلیل تجارت و مشخصات مورد نیاز ، طراحی ، برنامه نویسی و آزمایش ، موضوع قابل توجهی برای سالها بوده است. حتی امروز ، که رویه های مدیریت پروژه شرکت در حال بالغ شدن است ، طبق نتایج نظرسنجی موسسه مدیریت پروژه (PMI) Project Management Institute، فقط حدود نیمی از آنها (54٪) از اهمیت و ارزش این روش ها آگاه هستند.

فارغ از صنعت ، ثابت شده است که مدیریت پروژه یکی از عناصر مهم در کارایی یک شرکت و موفقیت نهایی آن است.

در واقع ، سازمان هایی که از شیوه های اثبات شده مدیریت پروژه استفاده می کنند 28 پول کمتری هدر می دهند و پروژه هایی را اجرا می کنند که 2.5 برابر موفقیت آمیزتر هستند.

متخصصان مدیریت پروژه نتیجه می گیرند که تعریف یک پروژه موفق تعریفی است که نه تنها در زمان مقرر و در حد بودجه تکمیل می شود ، بلکه منافع مورد انتظار را نیز به همراه دارد.

**مراحل مدیریت پروژه:**

صرف نظر از دامنه ، هر پروژه ای باید دنباله ای از اقدامات را کنترل و مدیریت کند.

طبق موسسه مدیریت پروژه (PMI) ، یک فرایند معمول مدیریت پروژه شامل مراحل زیر است:

1. شروع
2. برنامه ریزی
3. اجرا
4. عملکرد / نظارت
5. پروژه نزدیک است به پایان برسد یا خاتمه یافته است.

این مراحل به عنوان یک نقشه راه برای انجام وظایف خاص مورد استفاده قرار می گیرند ، چرخه حیات مدیریت پروژه را تعریف می کنند.

با این حال ، این ساختار بسیار کلی است.

یک پروژه معمولاً در هر مرحله دارای تعدادی مرحله داخلی است.

بسته به دامنه کار ، تیم ، صنعت و خود پروژه بسیار متفاوت هستند.

در تلاش برای یافتن یک رویکرد جهانی برای مدیریت هر پروژه ، بشریت تعداد قابل توجهی از تکنیک ها و روش های PM را توسعه داده است.

**روش های سنتی مدیریت پروژه:**

براساس چارچوب کلاسیک(سنتی) توضیح داده شده در بالا ، روش های سنتی رویکرد گام به گام اجرای پروژه را در پیش می گیرند.

بنابراین ، پروژه مراحل شروع ، برنامه ریزی ، اجرا ، نظارت مستقیماً به زمان بسته شدن در مراحل متوالی را طی می کند.

این رویکرد که غالباً خطی نامیده می شود ، شامل تعدادی از فازهای داخلی است که به ترتیب و به ترتیب زمانی اجرا می شوند.

معمولاً در صنعت ساخت و ساز و ساخت ، كه در هر مرحله تغییرات كمی لازم است و یا هیچ تغییری لازم نیست ، اعمال می شود ، مدیریت پروژه سنتی كاربرد خود را در مهندسی نرم افزار نیز پیدا كرده است.

این مدل که به عنوان مدل آبشار شناخته می شود ، از اوایل دهه 1970 که به طور رسمی توسط وینستون د. رویس توصیف شد ، یک روش غالب توسعه نرم افزار بوده است:

"دو مرحله اساسی وجود دارد که در همه پیشرفت های برنامه های رایانه ای مشترک است ، صرف نظر از اندازه و پیچیدگی.

در مرحله اول یک مرحله تجزیه و تحلیل وجود دارد ، دوم به دنبال یک مرحله کدگذاری انجام می شود ... این نوع مفهوم اجرای بسیار ساده در واقع تمام مواردی است که لازم است اگر تلاش به اندازه کافی کم باشد و اگر محصول نهایی توسط کسانی که آن را ساخته اند استفاده شود - معمولاً با برنامه های رایانه ای برای استفاده داخلی انجام می شود. "



**مدل آبشاری:**

****

**نگهداری**

**تایید**

**پیاده سازی**

**طراحی**

**مشخصات مورد نیاز**

**مدل آبشاری**

مدل آبشار تأکید زیادی بر برنامه ریزی و توسعه مشخصات دارد که حداکثر 40 درصد از زمان و بودجه پروژه را می گیرد.

اصل اساسی دیگر این روش ، نظم دقیق مراحل پروژه است.

مرحله پروژه جدید تا پایان مرحله قبلی آغاز نمی شود.

این روش برای پروژه های کاملاً مشخص با یک مهلت قابل تحویل و ثابت به خوبی کار می کند.

رویکرد آبشار نیاز به برنامه ریزی دقیق ، مستندات گسترده پروژه و کنترل دقیق روند توسعه دارد.

از نظر تئوری ، این امر باید منجر به تحویل به موقع بودجه ، ریسک کم پروژه و نتایج نهایی قابل پیش بینی شود.

با این حال ، هنگامی که به فرآیند مهندسی نرم افزار واقعی اعمال می شود ، روش آبشاری به دلیل محدودیت های متعدد ، کند ، پرهزینه و انعطاف پذیر است.

در بسیاری از موارد ، عدم توانایی آن در تطبیق محصول با نیازهای بازار در حال تکامل ، غالباً منجر به هدر رفت عظیم منابع و درنهایت شکست پروژه می شود.

**فلسفه مدیریت پروژه چابک**

بر خلاف رویکرد سنتی ، فلسفه مدیریت پروژه چابک به عنوان تلاشی برای انعطاف پذیری و کارایی بیشتر مهندسی نرم افزار معرفی شده است.

با داشتن 94 درصد از سازمانهای چابک در سال 2016 ، این استاندارد استاندارد صنعت مدیریت پروژه شده است.

تاریخچه چابکی را می توان در سال 1957 جستجو کرد:

در آن زمان برنی دیمسدیل ، جان فون نویمان ، هرب جیکوبز و جرالد وینبرگ از تکنیک های توسعه افزایشی (که امروزه با نام Agile شناخته می شوند) استفاده می کردند و نرم افزارهای ساخت IBM و Motorola را ساختند.

اگرچه ، نمی دانستند چگونه رویكردی را كه می پذیرند طبقه بندی كنند ، آنها به روشنی فهمیدند كه از بسیاری جهات با آبشار متفاوت است.

با این حال ، رویکرد چالاک امروزی به طور رسمی در سال 2001 ، هنگامی که گروهی متشکل از 17 متخصص توسعه نرم افزار برای بحث در مورد روش های جایگزین مدیریت پروژه ، ملاقات کردند ، مطرح شد.

آنها با داشتن دید روشنی از رویکرد توسعه نرم افزار انعطاف پذیر ، سبک و تیم محور ، آن را در مانیفست توسعه نرم افزار چابک ترسیم کردند.

مانیفست با هدف "کشف راه های بهتر تولید نرم افزار" ، اصول اساسی رویکرد جدید را مشخص می کند:

"از طریق این کار ما به ارزش رسیده ایم:

* افراد و تعاملات بیش از فرایندها و ابزارها
* نرم افزار کار بر روی اسناد جامع
* همکاری مشتری بر سر مذاکره در مورد قرارداد
* پاسخ به تغییر پیروی از یک برنامه.

این فلسفه که با دوازده اصل نرم افزار چابک تکمیل می شود ، راهی جدید و جهانی برای مدیریت پروژه ها است.

روش های چابک رویکرد تکراری برای توسعه نرم افزار دارند.

بر خلاف یک مدل آبشار خطی ساده ، پروژه های چابک از تعدادی چرخه کوچکتر - دو سرعت تشکیل شده اند.

هر یک از آنها یک پروژه در مینیاتور است:

این یک پرونده عقب مانده است و شامل مراحل طراحی ، اجرا ، آزمایش و استقرار در محدوده کاری از پیش تعریف شده است.



در پایان هر دو sprint ، یک محصول بالقوه قابل حمل تحویل می شود.

بنابراین ، با هر تکرار ویژگی های جدیدی به محصول اضافه می شود که نتیجه آن رشد تدریجی پروژه است.

با تأیید اعتبار ویژگی های ابتدای توسعه ، احتمال تحویل یک محصول بالقوه ناموفق به طور قابل توجهی کمتر است.

بیایید جنبه های اصلی چابکAgile را به طور خلاصه بیان کنیم:

* **Flexibility انعطاف پذیری:** دامنه کار ممکن است با توجه به نیازهای جدید تغییر کند.
* **Work breakdown خرابی کار:** این پروژه از چرخه های کوچک تشکیل شده است (معروف به Sprints in Scrum).
* **Value of teamwork ارزش کار تیمی:** اعضای تیم از نزدیک با هم کار می کنند و دید واضحی در مورد مسئولیت های خود دارند.
* **Iterative improvements:** ارزیابی مجدد مکرر کار انجام شده در یک چرخه برای بهتر شدن محصول نهایی انجام می شود.
* **Cooperation with a client** **همکاری با مشتری:** مشتری از نزدیک درگیر توسعه است و می تواند الزامات را تغییر دهد یا پیشنهادات تیم را بپذیرد.

با توجه به تحقیقات اخیر ، رویکرد چابک با اولویت دادن به انعطاف پذیری و چرخش سریع ، مزایای زیر را ارائه می دهد:

* توانایی مدیریت تغییر اولویت ها (88٪)
* افزایش بهره وری تیم از طریق تخصیص وظیفه روزانه (83٪)
* بهتر دیده شدن پروژه به دلیل سیستم برنامه ریزی ساده (83٪)

**چارچوب های چابک:**

چابک یک اصطلاح چتر برای طیف گسترده ای از روش ها و تکنیک ها است که اصول و ارزش های توضیح داده شده در بالا را به اشتراک می گذارد.

هر کدام از آنها زمینه های استفاده و ویژگی های متمایز خود را دارند. محبوب ترین چارچوب ها Scrum ، Kanban ، Hybrid ، Lean ، Bimodal و XP هستند. قبل از بحث در مورد این چارچوب ها با جزئیات بیشتر ، اجازه دهید ویژگی های اصلی آنها را بررسی کنیم.



**توضیحات شکل بالا:**

|  |  |
| --- | --- |
| **Planned mitigationتخفیف برنامه ریزی شده** | **frameworkچارچوب** |
| * **تمام دامنه کار به دوچرخه توسعه کوتاه تقسیم می شود.**
* **مدت sprints از یک تا چهار هفته است.**
* **تیم باید دقیقاً از یک برنامه کاری برای هر sprintاستفاده کند.**
* **افراد درگیر در یک پروژه نقش های از پیش تعیین شده ای دارند.**
 | **Scrum** |
| * **توسعه بر اساس تجسم گردش کار ساخته شده است.**
* **کار فعلی (کار در حال انجام یا WIP) در اولویت قرار دارد.**
* **هیچ دوره توسعه جعبه زمانی وجود ندارد.**
* **تیم می تواند برنامه کار را در هر زمان تغییر دهد.**
 | **Kanban** |
| * **چابک و آبشار مکمل یکدیگر هستند.**
* **توسعه نرم افزار چابک تحت شرایط آبشار (مهلت معین ، بودجه پیش بینی شده و از طریق ارزیابی ریسک) انجام می شود.**
 | **Hybrid**  |
| * **2 مدل جداگانه از کار سنتی (حالت 1) و چابک (حالت 2) وجود دارد.**
* **دو تیم جداگانه در حال کار بر روی پروژه هایی با 2 هدف متفاوت هستند.**
* **تیم حالت 1 زیرساختهای سیستم IT را حفظ می کند.**
* **تیم mode2 برنامه های ابتکاری ارائه می دهد.**
* **همکاری بین تیمی مهم است.**
 | **Bimodal**  |
| * **این چارچوب با تلاش ، وقت و هزینه کمتری به توسعه سریع نرم افزار کمک می کند.**
* **چرخه توسعه تا حد ممکن کوتاه است.**
* **محصول تحویل زودهنگام به طور مداوم در حال بهبود است.**
* **تیم مستقل است و مسئولیت های گسترده تری نسبت به مسئولیت های scrum ، bimodel و hybrid دارد.**
* **توسعه دهندگان همچنین می توانند مفهوم محصولات را فرموله کنند.**
 | **Lean**  |
| * **تمرکز جنبه های فنی توسعه نرم افزار است.**
* **XP شیوه های مهندسی را با هدف کمک به توسعه دهندگان برای نوشتن کد روشن معرفی می کند.**
* **توسعه محصول شامل مراحل مداوم است: نوشتن ، آزمایش ، تجزیه و تحلیل ، طراحی و ادغام مداوم کد.**
* **ارتباط چهره به چهره در تیم و مشارکت مشتری در توسعه بسیار مهم است.**
 | **Xp**  |

**Scrum: Roles, Sprints, and Artifacts اسکرام، نقش ها (قوانین) ، اسپرینت ها و مصنوعات (مشاهدات):**

اسکرام یک روش چابک غالب است.

این ماده به طور انحصاری توسط 58 درصد سازمان ها مورد استفاده قرار می گیرد در حالی که 18 درصد دیگر شرکت ها آن را با تکنیک های دیگر ترکیب می کنند. اولین بار در سال 1986 توسط Hirotaka Takeuchi و Ikujiro Nonaka در بازی توسعه محصول جدید توصیف شد ، تقریباً یک دهه بعد فرموله شد.

در سال 1995 ، کن شوابر و جف ساترلند ، نویسندگان کتاب راهنمای اسکرام ، آن را در کنفرانس OOPSLA ارائه دادند.

این ارائه بر اساس دانش آنها در حین استفاده از روش طی چند سال گذشته بود.

در حالی که اسکرام بسیار پیش از مانیفست چابک معرفی شده بود ، اما متکی به اصول چابک است و با ارزش های بیان شده در آن سند مطابقت دارد.

اسکرام با هدف حفظ همکاری قوی بین افرادی که روی محصولات پیچیده کار می کنند ، انجام می شود و جزئیات در حال تغییر یا اضافه شدن هستند.

این بر اساس تعاملات سیستماتیک بین سه نقش اصلی است: Scrum Master, Product Owner, and the Team.

* **Scrum Master**: یک شخصیت اصلی در یک پروژه است. مسئولیت اصلی او از بین بردن تمام موانعی است که ممکن است از کار موثر تیم جلوگیری کند.
* **Product Owner: معمولاً مشتری یا طرف دیگر ذینفع ، به طور فعال درگیر پروژه است ، دید جهانی محصول را منتقل می کند و بازخورد به موقع درباره کار انجام شده پس از هر سرعت را ارائه می دهد.**
* **Team :** گروهی متشکل از عملکرد و خود سازماندهی از افراد است که مسئولیت اجرای محصول را بر عهده دارند. برای انعطاف پذیری و بهره وری بیشتر باید از 7 عضو تیم تشکیل شود.

**Sprints and Artifacts:**

یک واحد اساسی کار در اسکرام - سرعت - یک چرخه توسعه کوتاه مدت است که برای تولید افزایش محصول قابل حمل مورد نیاز است.

طول sprint معمولاً بین 1 تا 4 هفته است: تکرارهای طولانی تر فاقد قابلیت پیش بینی و انعطاف پذیری هستند که از مزایای اساسی اسکرام هستند.

فاقد مدت زمان استاندارد (به شرطی که کمتر از 4 هفته باشد) ، تمام اسپرینتهای یک پروژه باید یک طول ثابت داشته باشند.

این امر برنامه ریزی و پیگیری پیشرفت را آسان می کند.



اسکرام به سه مصنوعات اصلی متکی است که برای مدیریت نیازها و ردیابی پیشرفت مورد استفاده قرار می گیرند - تعداد محصول عقب مانده ، عقب مانده اسپرینت ، نمودار برجسته Sprint.

این فرایند از طریق تعدادی از جلسات تکراری مانند Daily Scrum (Standup) ، Sprint Planning ، Review و Retrospective جلسات رسمی می شود.

**The product backlog:**

لیستی مرتب شده از موارد ویژگی است که ممکن است در محصول نهایی پروژه مورد نیاز باشد. این منبع واحد مورد نیاز است.

بک لاگ محصول به عنوان نیازهای جدید ، اصلاحات ، ویژگی ها و جزئیات در حال تغییر یا اضافه شدن است.

**The Sprint Backlog**:

لیستی از وظایفی است که تیم باید برای ارائه یک نرم افزار کاربردی در انتهای هر Sprint انجام دهد. به عبارت دیگر ، اعضای تیم در مورد اینکه کدام کالاها را تحویل می دهند توافق می کنند و طرحی را برای نحوه انجام آن تعریف می کنند.

**The Sprint Burndown Chart**:

تصویری از کار باقی مانده در اسپرینت است.

هم به تیم و هم به اسکرام مستر کمک می کند زیرا پیشرفت روز به روز را نشان می دهد و می تواند پیش بینی کند که آیا هدف اسپرینت طبق برنامه به دست خواهد آمد.

**Scrum Meetings:**

این فرایند از طریق تعدادی از جلسات تکرار شونده مانند Daily Scrum (Standup) ، برنامه Sprint Planning ، Review و Retrospective (Retrospective Sprint) رسمیت می یابد.

Daily Scrum جلسه ای با جدول زمانی است که طی آن یک تیم توسعه کار خود را هماهنگ کرده و برنامه ای برای 24 ساعت آینده تنظیم می کند.

این رویداد 15 دقیقه طول می کشد و باید روزانه در همان مکان و زمان برگزار شود.

کارهایی که باید تکمیل شود در Sprint Planning برنامه ریزی شده است.

همه افرادی که در Sprint شرکت دارند (صاحب محصول ، استاد Scrum ، و یک تیم توسعه) در این رویداد شرکت می کنند.

آنها به دو سوال اصلی پاسخ می دهند: کدام کار می تواند انجام شود و این کار چگونه انجام می شود.

برنامه ریزی حداکثر سرعت دویدن بیش از هشت ساعت برای یک ماه حداکثر سرعت دوام ندارد. برای اسپرینت های کوتاه تر ، جلسه معمولاً زمان کمتری طول می کشد.

در پایان هر Sprint ، تیم و صاحب محصول در Sprint Review ملاقات می کنند.

در طی این جلسه غیر رسمی ، تیم کار انجام شده را نشان می دهد و به سوالات مربوط به افزایش محصول پاسخ می دهد.

همه شرکت کنندگان برای افزایش ارزش محصول با هم همکاری می کنند که در مرحله بعدی چه کاری باید انجام دهند.

Sprint Review یک جلسه چهار ساعته با جعبه زمان برای Sprint های یک ماهه است.

کل تیم به جلسه های گذشته نگر می روند تا در مورد کار خود در دوره Sprint تأمل کنند.

شرکت کنندگان در مورد آنچه خوب یا اشتباه پیش رفته است بحث می کنند ، راه هایی برای بهبود پیدا می کنند و نحوه اجرای این تغییرات مثبت را برنامه ریزی می کنند.

Sprint Retrospective پس از بررسی و قبل از برنامه ریزی Sprint بعدی برگزار می شود. مدت زمان این رویداد برای Sprint های یک ماهه سه ساعت است.

**زمان استفاده از اسکرام**

اسکرام برای پروژه های پیچیده و طولانی مدت که به بازخورد ذینفعان احتیاج دارند ، به خوبی کار می کند که ممکن است تأثیر زیادی بر نیازهای پروژه بگذارد.

بنابراین ، وقتی نمی توان مقدار دقیق کار را تخمین زد و تاریخ انتشار آن ثابت نیست ، ممکن است Scrum بهترین انتخاب باشد.

با تعیین نیازهای مشتری و تحویل به موقع / بودجه به عنوان بالاترین اولویت ، اسکرام اعتماد 89 درصد از کاربران Agile را جلب کرده است.

بنابراین ، لیست شرکتهایی که از این روش استفاده می کنند چشمگیر است. در حقیقت ، یک صفحه گسترده عمومی با چنین سازمان هایی از جمله مایکروسافت ، IBM ، یاهو و گوگل وجود دارد.

آخرین تحقیقات Scrum Alliance حاکی از آن است که Scrum فراتر از IT است.

شرکت هایی که در زمینه های مالی ، مشاوره ای ، آموزشی ، خرده فروشی ، رسانه ای و سرگرمی فعالیت می کنند ، این روش را برای سازماندهی فرایندهای کاری خود و تقویت همکاری با مشتریان انتخاب می کنند.

در سال 2016 ، اکثر پاسخ دهندگان گزارش ایالت اسکرام (98 درصد) گفتند که برای حرکت به جلو از این چارچوب استفاده می کنند.

**Kanban: راه حل جامع برای رسیدگی به کارهای در حال انجام**

یکی دیگر از چارچوب های متداول مدیریت پروژه ، Kanban است.

چهل و سه درصد شرکت ها اظهار داشتند که از Kanban به عنوان یکی از چارچوب های مدیریت پروژه استفاده می کنند.

Kanban که از سیستم بصری کارتهایی که در ساخت تویوتا به عنوان یک روش کنترل تولید استفاده می شود ، رویکرد ساده و در عین حال قدرتمندی برای تولید محصولات نرم افزاری است.



Kanban به عنوان سیگنال بصری از ژاپنی ترجمه شده است ، بر روی تجسم گردش کار متمرکز است و کارهای در دست انجام (WIP) را در اولویت قرار می دهد و دامنه آن را محدود می کند تا با ظرفیت تیم مطابقت داشته باشد.

به محض انجام یک کار ، تیم می تواند مورد بعدی را از خط لوله بردارد.

بنابراین ، روند توسعه انعطاف پذیری بیشتری در برنامه ریزی ، چرخش سریع تر ، اهداف روشن و شفافیت ارائه می دهد.

برخلاف Scrum ، هیچ روشی استاندارد در این فرآیند و همچنین تکرارهای ثابت در Kanban مورد نیاز نیست.

توسعه پروژه براساس تجسم گردش کار از طریق یک صفحه Kanban انجام می شود که معمولاً با یادداشت های چسبناک و تخته های سفید یا ابزارهای آنلاین مانند Trello نشان داده می شود.



Trello Kanban را خودکار و دیجیتالی می کند.

با توجه به اطلاعات مختصر در مورد یک مورد کاری که هر کارت Kanban در آن قرار دارد ، همه اعضای تیم می دانند چه کسی مسئول این مورد است ، وظیفه هر شخص چیست ، چه زمانی قرار است آن را تمام کند و غیره

اعضای تیم همچنین می توانند نظرات خود را ارائه دهند ، تصاویر صفحه ، اسناد یا پیوندها را برای ارائه جزئیات بیشتر ضمیمه کنند.

تیم هایی که از ابزار Kanban استفاده می کنند به صورت همکاری کار می کنند.

توانایی ردیابی پیشرفت به همکاران کمک می کند تا سهم شخصی افراد را در دستیابی به هدف مشترک درک کنند و نتیجه آن تمرکز بر انجام به موقع کار باشد.

**چه زمانی باید از Kanban استفاده کرد؟**

با استفاده از Kanban ، تیم ها می توانند نسخه های کوچک منتشر کنند و با تغییر اولویت ها سازگار شوند.

بر خلاف اسکرام ، هیچ دو سرعت با اهداف از پیش تعیین شده آنها وجود ندارد.

kanban هنگام بروز کارهای متمرکز بر انجام کارهای کوچک است.

به عنوان مثال ، اگر آزمایش کنندگان خطایی در محصول پیدا کنند ، توسعه دهندگان سعی می کنند بلافاصله آنها را برطرف کنند.

به عنوان مثال ، Kanban پس از عرضه اصلی محصول به خوبی کار می کند.

شرکت هایی مانند Spotify و Wooga (شرکت پیشرو در ساخت بازی های موبایل) طی سال های گذشته از این رویکرد با موفقیت استفاده کرده اند. با این حال ، 8 درصد از سازمانها اسکرام را با تکنیک های kanban ترکیب می کنند و به اصطلاح از Scrumban استفاده می کنند تا چارچوب های اصلی.

**Hybrid ترکیبی:**

ترکیبی از آبشار و چابک (توسعه انعطاف پذیر و برنامه ریزی دقیق پروژه).

Agile و Waterfall دو دیدگاه متفاوت در زمینه مدیریت توسعه نرم افزار هستند.

مورد اول در مورد توسعه تکراری و انعطاف پذیر بودن است ، در حالی که مورد دوم ، توسعه گام به گام را ترویج می کند ، نیاز به برنامه ریزی دقیق دارد و ایجاد تغییرات در طول مسیر را رد می کند.

بیست و سه درصد شرکت ها دریافتند که استفاده از اصول هر دو روش می تواند سودمندتر از انتخاب یکی از دو روش باشد.

ترکیبی از رویکرد مدیریت پروژه آبشار سنتی و Agile ترکیبی نامیده می شود.

متخصصان از مزایای فلسفه Agile برای توسعه نرم افزار استفاده می کنند.

وقتی صحبت از بودجه بندی ، برنامه ریزی و تنظیم سخت افزار می شود ، Waterfall به خوبی کار می کند.

از طرف دیگر ، با تعبیه روشهای Agile در یک روند کار سنتی آبشار ، شرکت ها می توانند شانس تحقق پروژه های موفق را افزایش دهند.

به عنوان مثال ، برنامه ریزی پروژه می تواند در مسابقات سرعت انجام شود ، آزمایش می تواند در توسعه گنجانده شود و بازخورد به طور منظم جمع آوری شود.

روش های دیگر اصلاح مدل آبشار شامل استفاده از تخته های Kanban و سازماندهی گذشته نگرها است.

لازم به ذکر است که انتخاب ویژگی های چارچوب ترکیبی ممکن است به پروژه بستگی داشته باشد.

چارچوب های ترکیبی نه تنها به معنی استفاده از هر دو روش ، بسته به مرحله پروژه است ، بلکه شامل گزینه هایی برای تزریق روش های چابک به یک فرایند آبشار است.

**چه زمانی باید از Hybrid استفاده شود؟**

ترکیبی هنگامی که تحویل محصول به عملکرد سخت افزاری و نرم افزاری متکی باشد ، یک راه حل موثر است.

اما ، دلیل دیگری نیز برای انتخاب ترکیبی وجود دارد.

شرایطی که مشتری از یک بازه زمانی و بودجه مشخص نشده و همچنین عدم برنامه ریزی راضی نباشد ، نادر نیست.

چنین عدم اطمینان برای چابک معمول است.

در این حالت ، برنامه ریزی ، مشخصات مورد نیاز و طراحی برنامه می تواند در Waterfall انجام شود. Agile برای توسعه و آزمایش نرم افزار آماده است.