

## فصل اول

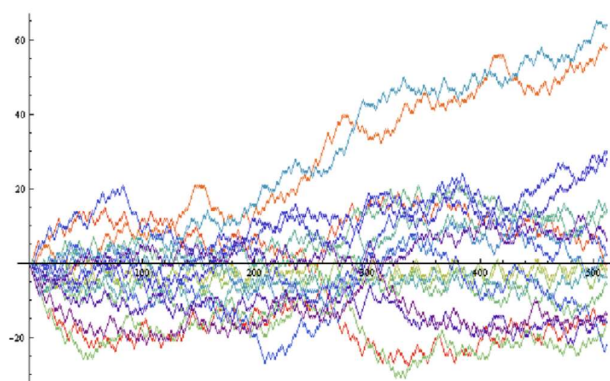
### ۱. مقدمه

## ۱-۱ مقدمه

تاریخچه استفاده از نمودارها به دوران باستان بازمی‌گردد، زمانی که انسان‌ها برای درک و نمایش داده‌ها و مفاهیم مختلف به استفاده از تصاویر و نمادها روی آوردند. در طول قرون وسطی و رنسانس، استفاده از نمودارها به ویژه در زمینه‌های نجوم، جغرافیا، و ریاضیات پیشرفت چشمگیری داشت. با شروع انقلاب صنعتی در قرن هجدهم، نمودارها به ابزارهای مهمی در مهندسی، اقتصاد، و علوم تبدیل شدند. توسعه نمودارهای مدرن مانند نمودارهای خطی، ستونی، و دایره‌ای در قرن نوزدهم توسط افرادی مانند ویلیام هنری پلیفر<sup>۳</sup> به استاندارد در تجزیه و تحلیل داده‌ها و نمایش تصویری اطلاعات تبدیل شد. [۱]

نمودارها در مهندسی نقش بی‌بدیلی دارند، زیرا امکان تجسم داده‌ها و تحلیل مفاهیم پیچیده را به صورت بصری فراهم می‌کنند. با استفاده از نمودارها، مهندسان قادرند تا الگوهای پنهان در داده‌ها را شناسایی کرده، روابط میان متغیرها را درک کنند و نتایج تحلیل‌ها را به طور موثر به دیگران منتقل نمایند. این ابزارهای گرافیکی، از طراحی سیستم‌های پیچیده تا مدیریت پروژه‌های بزرگ، در تمامی مراحل مهندسی کاربرد دارند. به ویژه در فرآیند طراحی و توسعه، نمودارها به مهندسان کمک می‌کنند تا ایده‌ها را به مدل‌های عملیاتی تبدیل کرده و از طریق شبیه‌سازی و تجزیه و تحلیل، بهینه‌سازی‌های لازم را انجام دهند. این امر منجر به افزایش دقت، کاهش خطا و تسهیل تصمیم‌گیری در پروژه‌های مهندسی می‌شود.

ترسیم نمودارها به وسیله انسان، به ویژه زمانی که با داده‌ها و سیستم‌های پیچیده سروکار داریم، می‌تواند فرآیندی چالش‌برانگیز و زمان‌بر باشد. دقت در ترسیم روابط پیچیده و نمایش صحیح داده‌ها نیازمند مهارت بالا و توجه دقیق به جزئیات است. هر گونه اشتباه در این فرآیند ممکن است به برداشت‌های نادرست و تصمیم‌گیری‌های اشتباه منجر شود. علاوه بر این، حجم بالای داده‌ها و نیاز به به‌روزرسانی مداوم نمودارها می‌تواند این کار را بسیار طاقت‌فرسا کند. از سوی دیگر، در برخی حوزه‌های مهندسی، نمودارها باید نه تنها دقیق بلکه کاملاً متناسب با استانداردهای تخصصی باشند که این امر نیاز به آشنایی کامل با قوانین و اصول طراحی دارد. به همین دلیل، ترسیم دستی نمودارها به خصوص در پروژه‌های بزرگ و پیچیده، علاوه بر خطاهای احتمالی، می‌تواند زمان و منابع زیادی را مصرف کند.



تصویر ۱-۱ نمونه ای از یک نمودار پیچیده

این نرم‌افزار با استفاده از متدولوژی چابک پیاده‌سازی شده است که باعث شده فرآیند توسعه آن به صورت تدریجی و انعطاف‌پذیر انجام گیرد. در این روش، پروژه به تعدادی چرخه یا چرخه کاری کوتاه تقسیم می‌شود که هر یک با هدف ارائه یک نسخه قابل استفاده از نرم‌افزار به پایان می‌رسد. این رویکرد امکان می‌دهد تا تیم توسعه به سرعت به بازخوردهای کاربران پاسخ

<sup>۳</sup> William Henry Playfair

دهد و به بهبود مداوم نرم‌افزار بپردازد. همچنین، استفاده از چابک تضمین می‌کند که نرم‌افزار با نیازهای متغیر کاربران و تغییرات محیط کسب‌وکار هماهنگ باشد و به‌روزرسانی‌های منظم و موثر انجام گیرد.

به منظور تضمین امنیت و محافظت از اطلاعات از الگوی احراز هویت در پروژه استفاده شده است. احراز هویت به این معناست که تنها کاربران مجاز به دسترسی به روابط از پیش‌نوشته‌شده‌ی خود هستند و از دسترسی غیرمجاز آنها به روابط دیگران جلوگیری شود. این فرآیند نه تنها برای حفاظت از اطلاعات ضروری است، بلکه به ایجاد اعتماد میان کاربران و سیستم نیز کمک می‌کند.

هدف ما در این پروژه، ساخت یک نرم‌افزار ترسیم نمودار با استفاده از استانداردهای مهندسی نرم‌افزار و روش چابک است. اینکه بتوانیم با استفاده از چارچوب مطرح شده یک نرم‌افزار ترسیم نمودار طراحی و پیاده‌سازی کنیم و میزان هزینه‌ی آن را با زمانی که از این الگو پیروی نکنیم مقایسه کنیم.

در همین راستا در این گزارش، ابتدا بعد از تعریف مفاهیم اولیه‌ی به کار رفته از مهندسی نرم‌افزار در این پروژه، نگاهی اجمالی به ساختار چرخه‌های کاری می‌اندازیم. بعد از آن به بررسی هر یک از این چرخه‌ها پرداخته و در نهایت، عملکرد سیستم و بهره‌وری آن را بررسی می‌کنیم.

## فصل دوم

### ۲. روش چابک

## ۲-۱ تعریف روش چابک

اصل چابک<sup>۴</sup> یک متدولوژی مدیریت پروژه و توسعه نرم‌افزار است که بر انعطاف‌پذیری، همکاری تیمی، و پاسخگویی سریع به تغییرات تأکید دارد. این رویکرد برخلاف روش‌های سنتی، که معمولاً بر یک برنامه‌ریزی جامع و تغییرناپذیر در ابتدای پروژه تکیه می‌کنند، اجازه می‌دهد تا پروژه‌ها به‌صورت تدریجی و از طریق چرخه‌های کوتاه و تکراری که به آن‌ها چرخه کاری<sup>۵</sup> گفته می‌شود، پیشرفت کنند. در هر چرخه، یک بخش مشخص از پروژه تکمیل می‌شود و محصولی قابل ارائه و استفاده به دست می‌آید که می‌تواند بازخوردهای کاربران را در خود جای دهد. این فرآیند به تیم‌ها امکان می‌دهد تا به سرعت به نیازها و اولویت‌های متغیر کاربران پاسخ دهند و نرم‌افزار یا محصول نهایی به‌طور مستمر بهبود یابد.

علاوه بر این، اصل چابک بر ارتباطات مستمر و نزدیک بین اعضای تیم و ذی‌نفعان تأکید دارد. در این رویکرد، جلسات منظم برای بازبینی عملکرد، ارزیابی پیشرفت، و تعیین اهداف جدید برگزار می‌شود. همچنین، تیم‌ها تشویق می‌شوند تا خودسازمان‌دهی داشته باشند و تصمیمات را به صورت مشارکتی و مبتنی بر بازخوردهای جمعی اتخاذ کنند. این شیوه کاری باعث می‌شود تا نه تنها محصول نهایی با کیفیت بالاتر و مطابق با نیازهای واقعی کاربران باشد، بلکه محیط کاری پویاتر و خلاقانه‌تری نیز ایجاد گردد. به طور کلی، اصل چابک با تأکید بر سرعت، انعطاف‌پذیری، و همکاری، به سازمان‌ها کمک می‌کند تا در مواجهه با تغییرات و چالش‌ها، کارآمدتر و موفق‌تر عمل کنند.

در جلسات روش چابک با ذی‌نفعان، چندین موضوع کلیدی مورد بحث قرار می‌گیرد که هدف اصلی آن‌ها اطمینان از هماهنگی، شفافیت و تطابق پروژه با نیازهای کاربران و کسب‌وکار است. این جلسات معمولاً به صورت منظم و در فواصل زمانی کوتاه، مانند پایان هر چرخه کاری برگزار می‌شوند و به عنوان بخش مهمی از فرآیند توسعه چابک شناخته می‌شوند. برخی از موضوعات اصلی که در این جلسات مورد بحث قرار می‌گیرد عبارتند از:

۱. بررسی پیشرفت پروژه: تیم توسعه گزارش می‌دهد که در طول چرخه کاری گذشته چه میزان پیشرفت حاصل شده است و چه ویژگی‌ها یا عملکردهایی به نرم‌افزار اضافه شده‌اند. این بخش از جلسه به ذی‌نفعان کمک می‌کند تا وضعیت فعلی پروژه را به‌خوبی درک کنند.
۲. پیش‌نمایش محصول: یکی از بخش‌های مهم جلسات با ذی‌نفعان، ارائه یک پیش‌نمایش زنده از قابلیت‌ها و ویژگی‌های جدید توسعه یافته در طول چرخه کاری است. این پیش‌نمایش محصول به ذی‌نفعان این امکان را می‌دهد تا عملکرد واقعی نرم‌افزار را ببینند و بازخورد فوری ارائه دهند. در ادامه برخی از روش‌های پیش‌نمایش را توضیح می‌دهیم.
۳. بازخورد و بررسی: ذی‌نفعان نظرات و پیشنهادات خود را درباره ویژگی‌های ارائه شده و جهت کلی پروژه بیان می‌کنند. این بازخوردها بسیار حیاتی هستند زیرا به تیم توسعه کمک می‌کنند تا در اسپرینت‌های بعدی به بهبود محصول و تطابق بهتر آن با نیازهای کاربران بپردازند.
۴. بررسی مشکلات و چالش‌ها: هرگونه مشکلات یا چالش‌هایی که در طول چرخه پیش آمده است، با ذی‌نفعان در میان گذاشته می‌شود. این می‌تواند شامل موانع فنی، مشکلات تیمی، یا تغییرات غیرمنتظره در نیازهای پروژه باشد.
۵. برنامه‌ریزی برای چرخه بعدی: در پایان جلسه، برنامه‌ها و اهداف چرخه بعدی با ذی‌نفعان به اشتراک گذاشته می‌شود. این شامل تعیین اولویت‌ها، ویژگی‌های مورد نظر، و زمان‌بندی است. هدف از این بخش اطمینان از هماهنگی کامل بین تیم توسعه و ذی‌نفعان در مورد مسیر پیش رو است.

<sup>۴</sup> Agile

<sup>۵</sup> Sprint

به‌طور کلی، جلسات با ذی‌نفعان در روش چابک یک فرصت عالی برای ایجاد شفافیت، ارتباط نزدیک‌تر، و تضمین اینکه پروژه در راستای اهداف کسب‌وکار و نیازهای کاربران پیش می‌رود، فراهم می‌کند.

پیش‌نمایش<sup>۶</sup> یکی از مهم‌ترین بخش‌های جلسات چابک است که در آن تیم توسعه نتایج کار خود را به ذی‌نفعان و دیگر اعضای تیم ارائه می‌دهد. روش‌های مختلفی برای انجام پیش‌نمایش وجود دارد که بسته به نوع پروژه، محصول و مخاطبان می‌تواند متفاوت باشد. در زیر برخی از روش‌های رایج پیش‌نمایش را توضیح می‌دهم:

#### ۱. پیش‌نمایش زنده<sup>۷</sup>

- توضیح: در این روش، تیم توسعه محصول یا ویژگی‌های جدید را به صورت زنده و در محیط واقعی به ذی‌نفعان نمایش می‌دهد. این روش معمولاً بیشترین تأثیر را دارد زیرا ذی‌نفعان می‌توانند به‌صورت مستقیم عملکرد و کارایی محصول را مشاهده کنند.

- مزایا: شفافیت بالا، امکان دریافت بازخورد فوری، و نمایش محصول در حالت واقعی.

- چالش‌ها: نیازمند آماده‌سازی دقیق و بدون خطا بودن اجرای زنده است که ممکن است استرس‌زا باشد.

#### ۲. پیش‌نمایش ویدئویی<sup>۸</sup>

- توضیح: در این روش، یک ویدئو از عملکرد محصول یا ویژگی‌های جدید از پیش ضبط می‌شود و در جلسه پیش‌نمایش به نمایش گذاشته می‌شود. این روش به ویژه زمانی مفید است که نمایش زنده به دلیل محدودیت‌های فنی یا زمانی امکان‌پذیر نباشد.

- مزایا: امکان ویرایش و آماده‌سازی دقیق ویدئو، مناسب برای نمایش‌های پیچیده یا زمان‌بندی محدود.

- چالش‌ها: فقدان امکان تعامل زنده و دریافت بازخورد فوری.

#### ۳. پیش‌نمایش تعاملی<sup>۹</sup>

- توضیح: در این روش، به ذی‌نفعان اجازه داده می‌شود که خودشان با محصول یا ویژگی‌های جدید به صورت مستقیم تعامل داشته باشند. این نوع پیش‌نمایش معمولاً به صورت یک محیط تستی یا نسخه نمایشی محصول انجام می‌شود که کاربران می‌توانند آن را امتحان کنند.

- مزایا: تجربه کاربری واقعی، امکان دریافت بازخورد عملی از کاربران، و ارتقای اعتماد به محصول.

- چالش‌ها: نیاز به محیط تستی قابل اعتماد و زمان بیشتر برای اجرای دمو.

#### ۴. پیش‌نمایش اسلایدی<sup>۱۰</sup>

- توضیح: این روش شامل استفاده از اسلایدها برای نمایش و توضیح ویژگی‌ها و عملکردهای جدید است. این روش بیشتر بر روی توضیحات مفهومی و چگونگی کارکرد محصول تمرکز دارد و کمتر به نمایش زنده یا تعاملی می‌پردازد.

<sup>۶</sup> Demo

<sup>۷</sup> Live Demo

<sup>۸</sup> Video Demo

<sup>۹</sup> Interactive Demo

<sup>۱۰</sup> Slide-Based Demo

- مزایا: ساده برای آماده‌سازی، مناسب برای توضیح ویژگی‌های مفهومی یا نظری، و مدیریت زمان بهتر.

- چالش‌ها: ممکن است نتواند تجربه واقعی استفاده از محصول را به خوبی منتقل کند.

۵. پیش‌نمایش از راه دور<sup>۱۱</sup>

- توضیح: این روش برای زمانی که ذی‌نفعان در مکان‌های مختلف جغرافیایی قرار دارند و امکان حضور فیزیکی در یک جلسه را ندارند، به کار می‌رود. معمولاً از طریق ابزارهای ویدئوکنفرانس یا اشتراک‌گذاری صفحه نمایش انجام می‌شود.

- مزایا: انعطاف‌پذیری مکانی، امکان ارائه به ذی‌نفعان در هر نقطه‌ای از جهان، و کاهش هزینه‌های سفر.

- چالش‌ها: وابستگی به کیفیت اینترنت و تجهیزات ارتباطی، و محدودیت‌های مربوط به تعامل زنده.

هر یک از این روش‌ها بسته به نیاز پروژه، سطح آمادگی محصول، و نوع ذی‌نفعان می‌تواند به صورت مستقل یا ترکیبی مورد استفاده قرار گیرد. انتخاب روش مناسب برای پیش‌نمایش می‌تواند تأثیر زیادی بر موفقیت جلسات و دریافت بازخوردهای ارزشمند داشته باشد.

## ۲-۲ تاریخچه‌ای کوتاه

تاریخچه روش چابک به‌عنوان یک متدولوژی نوین در توسعه نرم‌افزار و مدیریت پروژه، از نیاز به تغییر در رویکردهای سنتی و ناکارآمد در این حوزه‌ها سرچشمه می‌گیرد. برای درک کامل این تاریخچه، باید به چالش‌ها و محدودیت‌های روش‌های قدیمی در توسعه نرم‌افزار و چگونگی پیدایش و تکامل روش چابک بپردازیم.

پیش از ظهور روش چابک، بسیاری از پروژه‌های نرم‌افزاری بر اساس روش‌های سنتی مانند مدل آبشاری<sup>۱۲</sup> اجرا می‌شدند. در مدل آبشاری، فرآیند توسعه نرم‌افزار به مراحل مجزا و متوالی تقسیم می‌شود، از جمله نیازسنجی، طراحی، پیاده‌سازی، آزمایش و نگهداری. هر مرحله باید قبل از شروع مرحله بعدی به‌طور کامل تکمیل شود. این رویکرد به برنامه‌ریزی دقیق و جامع در ابتدای پروژه نیاز داشت و تغییرات در مراحل بعدی بسیار دشوار و پرهزینه بود.

با این حال، با گذشت زمان، مشکلات متعددی در اجرای این روش پدیدار شد. یکی از بزرگ‌ترین چالش‌ها این بود که نیازهای کاربران و بازار به سرعت تغییر می‌کرد و مدل آبشاری توانایی لازم برای پاسخگویی به این تغییرات را نداشت. علاوه بر این، پروژه‌ها اغلب با تأخیرهای زیاد و هزینه‌های بیشتر از حد پیش‌بینی مواجه می‌شدند، و در نهایت محصول نهایی ممکن بود با نیازهای واقعی کاربران تطابق نداشته باشد.

در دهه ۱۹۹۰، با توجه به مشکلات روش‌های سنتی، گروهی از متخصصان نرم‌افزار به دنبال راهکارهای جدیدی برای افزایش انعطاف‌پذیری و پاسخگویی در توسعه نرم‌افزار بودند. این تلاش‌ها به تدریج منجر به ظهور روش‌ها و چارچوب‌های جدیدی مانند اسکرام<sup>۱۳</sup> شد. این روش‌ها همگی به نوعی تلاش داشتند تا با ایجاد تغییرات کوچک و تدریجی در پروژه‌ها، به سرعت به نیازهای متغیر کاربران پاسخ دهند.

<sup>۱۱</sup> Remote Demo

<sup>۱۲</sup> Waterfall

<sup>۱۳</sup> Scrum

نقطه عطف واقعی در تاریخچه روش چابک در فوریه سال ۲۰۰۱ اتفاق افتاد، زمانی که ۱۷ نفر از متخصصان برجسته نرم‌افزار، گرد هم آمدند تا در مورد روش‌های جدید توسعه نرم‌افزار بحث و گفتگو کنند. این گروه شامل افرادی بود که هر کدام تجربه قابل توجهی در توسعه نرم‌افزار داشتند.

نتیجه این نشست، تدوین بیانیه چابک<sup>۱۴</sup> بود که چهار ارزش اصلی و دوازده اصل را به‌عنوان مبانی توسعه چابک نرم‌افزار معرفی کرد. این ارزش‌ها عبارت بودند از:

۱. افراد و تعاملات بیشتر از فرآیندها و ابزارها.

۲. نرم‌افزار کارآمد بیشتر از مستندات جامع.

۳. همکاری نزدیک با مشتری بیشتر از قراردادهای رسمی.

۴. پاسخگویی به تغییرات بیشتر از پیروی از یک برنامه‌ی از پیش تعیین شده.

این بیانیه به سرعت مورد استقبال جامعه نرم‌افزاری قرار گرفت و به‌عنوان یک استاندارد برای توسعه چابک نرم‌افزار شناخته شد.

پس از انتشار بیانیه چابک، روش‌ها و چارچوب‌های مختلف چابک به سرعت در سازمان‌ها و تیم‌های توسعه نرم‌افزار مورد استفاده قرار گرفتند. اسکرام و XP از جمله اولین و پرکاربردترین روش‌های چابک بودند که به دلیل ساختار ساده و قابلیت اجرای آسان، به سرعت محبوبیت پیدا کردند.

با گذشت زمان، روش چابک از حوزه توسعه نرم‌افزار فراتر رفت و به دیگر حوزه‌های مدیریت پروژه نیز وارد شد. شرکت‌ها و سازمان‌های مختلف به مزایای این روش، از جمله کاهش زمان تحویل، افزایش رضایت مشتری، و بهبود کیفیت محصول، پی بردند و آن را به کار گرفتند.

با وجود موفقیت گسترده، روش چابک نیز با چالش‌ها و انتقاداتی مواجه شده است. برخی از این چالش‌ها شامل نیاز به تغییر فرهنگ سازمانی، پیچیدگی در مقیاس‌دهی به پروژه‌های بزرگ، و نیاز به همکاری نزدیک‌تر و مداوم بین تیم‌ها و ذی‌نفعان است. با این حال، با گذشت زمان و تکامل روش‌های مختلف چابک، بسیاری از این چالش‌ها نیز مورد توجه قرار گرفته و راه‌حل‌های مناسبی برای آن‌ها ارائه شده است.

امروزه، روش چابک به یکی از رایج‌ترین و پرکاربردترین متدولوژی‌های مدیریت پروژه و توسعه نرم‌افزار تبدیل شده است. بسیاری از سازمان‌ها در سراسر جهان از این روش برای مدیریت پروژه‌های خود استفاده می‌کنند و در تلاش هستند تا با تطبیق آن با نیازهای خاص خود، از مزایای چابک بهره‌مند شوند. روش چابک به‌طور مداوم در حال تکامل است و با افزودن چارچوب‌ها و ابزارهای جدید، همچنان به‌عنوان یک رویکرد پویا و کارآمد در مدیریت پروژه‌ها مورد توجه قرار دارد.

<sup>۱۴</sup> Agile Manifesto



## ۲-۳ عملکرد

روش چابک بر اساس تکرارهای کوتاه‌مدت، همکاری نزدیک، و انعطاف‌پذیری در پاسخ به تغییرات بنا شده است. نحوه عملکرد این روش از طریق فرآیندهای مشخص، نقش‌های کلیدی، و ابزارهای گوناگون توضیح داده می‌شود. در ادامه، نحوه عملکرد روش چابک را به صورت مفصل توضیح خواهیم داد.

روش چابک بر اساس چهار ارزش اصلی و دوازده اصل کار می‌کند. این اصول به این معنی هستند که تیم‌های چابک باید:

- به تعاملات بین افراد بیشتر از فرآیندها و ابزارها اهمیت دهند.

- تمرکز بیشتری بر روی ارائه نرم‌افزار کارآمد نسبت به مستندات جامع داشته باشند.

- با مشتریان نزدیک‌تر همکاری کنند.

- به تغییرات به جای پیروی کورکورانه از برنامه‌های از پیش تعیین‌شده، به سرعت پاسخ دهند.

یکی از ویژگی‌های اصلی روش چابک، تقسیم پروژه به دوره‌های زمانی کوتاه و است. هر چرخه کاری معمولاً بین ۱ تا ۴ هفته طول می‌کشد و در این مدت، تیم توسعه بر روی یک مجموعه کوچک از ویژگی‌ها یا وظایف که به مجموعه چرخه<sup>۱۵</sup> معروف است، تمرکز می‌کند. در هر اسپرینت، هدف اصلی تولید یک محصول یا نسخه قابل ارائه<sup>۱۶</sup> است که قابل تست و ارزیابی باشد. چرخه‌ها به تیم‌ها امکان می‌دهند تا به صورت تکراری و تدریجی پیشرفت کنند و محصول نهایی را به تدریج بهبود بخشند.

روش چابک بر پایه برگزاری جلسات منظم برای هماهنگی و ارزیابی پیشرفت پروژه بنا شده است. این جلسات به طور معمول شامل موارد زیر هستند:

- جلسه برنامه‌ریزی چرخه کاری: در آغاز هر چرخه، تیم با کمک صاحب محصول (Product Owner) وظایف و اهداف آن چرخه کاری را تعیین می‌کند. در این جلسه، لیست نیازمندی‌های محصول بررسی می‌شود و تیم تصمیم می‌گیرد که چه وظایفی را در چرخه کاری پیش رو انجام دهد.

- جلسه روزانه (Daily Standup): هر روز، تیم یک جلسه کوتاه (معمولاً ۱۵ دقیقه) برگزار می‌کند که در آن هر عضو تیم گزارش می‌دهد که چه کاری انجام داده، چه کاری خواهد کرد، و چه مشکلاتی ممکن است وجود داشته باشد. این جلسات به هماهنگی روزانه تیم کمک می‌کند و موانع را به سرعت شناسایی می‌کند.

- جلسه بازنگری چرخه کاری (Sprint Review): در پایان هر اسپرینت، تیم نسخه جدید محصول را به ذی‌نفعان و صاحب محصول ارائه می‌دهد. در این جلسه، تیم بازخوردهای فوری دریافت می‌کند و عملکرد خود را ارزیابی می‌کند.

- جلسه بازنگری عملکرد (Sprint Retrospective): پس از بازنگری اسپرینت، تیم در یک جلسه داخلی شرکت می‌کند تا فرآیندهای خود را بررسی کند و برای بهبود عملکرد در اسپرینت‌های آینده تصمیم‌گیری کند. این جلسه بر روی چگونگی بهبود فرآیندها و همکاری تیمی تمرکز دارد.

چندین نقش کلیدی در تیم‌های چابک وجود دارد که هر کدام مسئولیت‌های مشخصی دارند:

<sup>۱۵</sup> Sprint Backlog

<sup>۱۶</sup> Increment

- صاحب محصول (Product Owner): مسئول تعریف و اولویت‌بندی ویژگی‌ها و نیازمندی‌های محصول است. صاحب محصول نماینده ذی‌نفعان و مشتریان است و باید اطمینان حاصل کند که تیم در مسیر تحقق اهداف کسب‌وکار قرار دارد.

- اسکرام مستر (Scrum Master): مسئول تسهیل فرآیندها و جلسات چابک است. اسکرام مستر به تیم کمک می‌کند تا بهترین عملکرد را داشته باشد و هرگونه مانعی که در راه است را برطرف می‌کند. اسکرام مستر همچنین از تطابق تیم با اصول چابک اطمینان حاصل می‌کند.

- تیم توسعه (Development Team): گروهی از افراد چندمهارته که به‌طور مشترک بر روی توسعه محصول کار می‌کنند. این تیم شامل برنامه‌نویسان، طراحان، تسترها و سایر افرادی است که برای تحویل نسخه‌های کاربردی محصول همکاری می‌کنند.

بک‌لاگ محصول (Product Backlog) لیستی از تمامی ویژگی‌ها، بهبودها، اصلاحات و نیازمندی‌هایی است که باید در محصول نهایی پیاده‌سازی شوند. این لیست توسط صاحب محصول نگهداری می‌شود و دائماً با توجه به بازخوردها و تغییرات به‌روزرسانی می‌شود.

بک‌لاگ چرخه کاری (Sprint Backlog) زیرمجموعه‌ای از بک‌لاگ محصول است که تیم توسعه تصمیم می‌گیرد در طول یک چرخه کاری خاص انجام دهد. این لیست معمولاً وظایف کوچک‌تر و دقیق‌تری را شامل می‌شود که تیم می‌تواند در مدت زمان کوتاه چرخه کاری تکمیل کند.

یکی از اصول مهم چابک، تحویل مستمر نسخه‌های قابل اجرا از محصول است. هر اسپرینت باید با یک نسخه جدید از محصول که قابل تست و استفاده است به پایان برسد. این رویکرد به تیم‌ها اجازه می‌دهد تا به سرعت بازخورد کاربران و ذی‌نفعان را دریافت کرده و در اسپرینت‌های بعدی بهبودهای لازم را اعمال کنند.

این بازخورد مداوم به اطمینان از اینکه محصول نهایی با نیازهای واقعی کاربران و تغییرات بازار تطابق دارد، کمک می‌کند. یکی از بزرگ‌ترین مزایای روش چابک، توانایی آن در مدیریت تغییرات است. در چابک، تغییرات به‌عنوان یک واقعیت اجتناب‌ناپذیر در فرآیند توسعه پذیرفته می‌شوند. تیم‌ها به‌جای مقابله با تغییرات، آن‌ها را به‌عنوان فرصتی برای بهبود محصول و رضایت مشتریان می‌بینند.

بک‌لاگ محصول و اولویت‌های آن به‌طور مداوم بر اساس بازخوردها و تغییرات محیطی به‌روزرسانی می‌شود. این انعطاف‌پذیری به تیم‌ها اجازه می‌دهد تا به سرعت به تغییرات پاسخ دهند و از انحراف پروژه از اهداف خود جلوگیری کنند.

روش چابک بر همکاری و ارتباطات نزدیک بین تمامی اعضای تیم و ذی‌نفعان تأکید دارد. این ارتباطات نزدیک به شفافیت در فرآیند توسعه، سرعت در شناسایی مشکلات، و بهبود کارایی تیم کمک می‌کند.

علاوه بر جلسات رسمی، تیم‌های چابک تشویق می‌شوند تا به‌صورت غیررسمی و در محیط کاری خود نیز به‌طور مستمر با هم در ارتباط باشند. این محیط همکاری باز به تیم‌ها اجازه می‌دهد تا سریع‌تر تصمیم‌گیری کنند و مشکلات را به‌صورت جمعی حل کنند.

روش چابک یک فرآیند ارزیابی و بهبود مستمر را دنبال می‌کند. تیم‌ها نه تنها به‌صورت مداوم عملکرد محصول را ارزیابی می‌کنند، بلکه فرآیندهای کاری خود را نیز مورد بازبینی قرار می‌دهند تا بهره‌وری و کیفیت کار را بهبود بخشند.

جلسات بازنگری اسپرینت و بازنگری عملکرد از جمله ابزارهایی هستند که به تیم‌ها کمک می‌کنند تا به این هدف دست یابند. تیم‌ها با تحلیل عملکرد گذشته، می‌توانند از تجربیات خود برای بهبود فرآیندهای آتی استفاده کنند.

## نتیجه‌گیری

روش چابک علی‌رغم روش‌های قبلی خود بسیار پویا بوده و به دلیل توجه بیشتر بین تعاملات افراد، سرعت انجام کار را بالا می‌برد. همچنین با ارائه‌ی نسخه‌های متعدد و در بازه‌های زمانی کوتاه، به ذی‌نفعان و مشتری این دلگرمی را می‌دهد که به محصول نهایی خود هر روز نزدیکتر می‌شوند. همچنین بدلیل جلسات کوتاه روزانه، کل تیم از عملکرد دیگران مطلع شده و هماهنگی بین اعضا را بالا می‌برد.