

برديا نيك بخش

۹۹۱۴۳۵۵۱۵۰

V Model

in Software Engineering

مدل V یک رویکرد ساختاریافته برای توسعه و تست پروژه‌ها است که روی تست در مراحل مختلف توسعه تاکید دارد. این نمونه یکی از انواع مدل‌های توسعه نرم افزار بوده که با تعیین اهداف روشن برای پروژه شروع می‌شود و سپس کار را به سیستم‌های کوچک‌تر و اجزای جداگانه تقسیم می‌کند. در هر مرحله از مدل وی (V model)، یک تست مرتبط وجود دارد تا اطمینان حاصل شود که همه چیز همانطور که در نظر گرفته شده، عمل می‌کند.

### مراحل صحت‌سنجی (Verification) مدل وی (V model)

در مراحل صحت‌سنجی، تیم‌های توسعه بدون اجرای کد، توسعه محصول را ارزیابی و بررسی می‌کنند. این کار به آن‌ها کمک می‌کند تا تعیین کنند که آیا محصول نهایی می‌تواند تمام نیازهای مشتری را برآورده کند؟ در ادامه مراحل اعتبارسنجی و تایید مدل V را بررسی می‌کنیم.

### تجزیه و تحلیل نیازمندی‌های کسب و کار (Business requirement analysis)

تحلیل نیازمندی‌های کسب و کار اولین مرحله در چرخه عمر توسعه نرم افزار است. در طول این مرحله، مدیر پروژه با مشتری‌ها ارتباط برقرار می‌کند تا نیازها و انتظارات آن‌ها را درک کند. این کار به مدیر پروژه اجازه می‌دهد تا نیازهای دقیق یک پروژه را تعیین کند. مدیر پروژه از این اطلاعات برای ایجاد یک طرح تست پذیرش استفاده می‌کند.

در واقع مدیر پروژه فهرستی از تمام نیازهای مشتری ایجاد می‌کند تا تیم، مجموعه‌ای واضح از دستورالعمل‌ها را در اختیار داشته باشد و دنبال کند. تیم توسعه همچنین تست‌های پذیرش کاربر (UAT) را ایجاد می‌کنند تا بتوانند تجربه کاربر را بعداً در مرحله تایید اعتبار بسنجند.

### طراحی سیستم (System design)

در طول مرحله طراحی سیستم، تیم کاربرد عملی پروژه‌ای را که در حال توسعه است، شناسایی می‌کند. این مرحله، شناسایی عملکردهای مختلفی که پروژه باید در نهایت انجام دهد و اینکه چگونه آن عملکردها با یکدیگر یا با عناصر خارجی تعامل دارند را شامل می‌شود.

در طول این مرحله، شرکت همچنین آزمایش‌های سیستمی را توسعه می‌دهد که باعث می‌شود از عملکرد صحیح کل سیستم اطمینان حاصل شود و تیم می‌تواند در مراحل بعدی توسعه، این تست‌ها را انجام دهد.

### طراحی معماری (Architectural design)

در مرحله طراحی معماری، تیم توسعه، زیرساخت‌های محصول را برنامه‌ریزی می‌کند. در طول این مرحله، توسعه‌دهنده‌ها اجزای پروژه و نحوه ارتباط و تعامل آن‌ها با یکدیگر را تعریف می‌کنند. این کار چارچوبی را برای درک نحوه عملکرد پروژه پس از توسعه نهایی به وجود می‌آورد.

توسعه‌دهنده‌ها همچنین تست‌های یکپارچه‌سازی را در این مرحله انجام می‌دهند و آزمایش‌هایی را مشخص می‌کنند که به تیم اجازه می‌دهد تا اطمینان حاصل کند که تمام اجزای سیستم در پروژه می‌توانند به درستی با یکدیگر ارتباط برقرار کرده و تعامل داشته باشند.

### طراحی ماژول (Module design)

در طول مرحله طراحی ماژول، تیم یک طراحی داخلی دقیق برای هر ماژول سیستم ایجاد می‌کند. اغلب از طراحی ماژول، به عنوان طراحی سطح پایین یاد می‌شود. این مرحله شامل تجزیه سیستم به واحدهای کوچکتری است که ساخت آن‌ها آسان‌تر است. سند طراحی ماژول تمام جداول پایگاه داده، جزئیات رابط کاربری، مسائل مربوط به وابستگی، لیست پیام‌های خطا، ورودی‌ها و خروجی‌ها را مشخص می‌کند تا برنامه‌نویس‌ها بتوانند کدنویسی هر ماژول را آغاز کنند.

تیم‌ها همچنین طرح‌های تست واحد (UTPs) را در مرحله طراحی ماژول ایجاد می‌کنند تا بتوانند عملکرد هر ماژول را بعداً در مرحله اعتبارسنجی تست واحد، ارزیابی کنند.

### پیاده‌سازی (Implementation)

در مرحله پیاده‌سازی، برنامه‌نویس‌ها بر اساس سیستم و الزامات طراحی معماری، زبان برنامه‌نویسی را انتخاب می‌کنند. سپس، شروع به کدنویسی هر ماژول می‌کنند. آن‌ها همچنین کدها را تست می‌کنند و در صورت نیاز آن‌ها را برای بهبود عملکرد بهینه می‌کنند. زمانی که یک ماژول تکمیل شد، می‌توانند آن را به مخزن (Repository) اضافه کنند Repository .. یک مکان مرکزی ذخیره‌سازی فایل است که همه فایل‌های کد منبع یک برنامه خاص را در خود جای می‌دهد.

### مراحل اعتبارسنجی (Validation) مدل وی (V model)

در مراحل اعتبارسنجی، تیم کد را برای انجام تست‌های عملکردی و غیرعملکردی اجرا می‌کند. این کار به آن‌ها کمک می‌کند تا نرم‌افزار را پس از توسعه ارزیابی و اطمینان حاصل کنند که محصول نهایی نیازهای مشتری را برآورده می‌کند. در ادامه مراحل اعتبارسنجی را در مدل V بررسی می‌کنیم.

## تست واحد (Unit testing)

در طول مرحله تست واحد، تیم از طرح‌های تست واحد یا همان UTP های توسعه یافته در مرحله طراحی مازول برای آزمایش کد، شناسایی مشکلات احتمالی و توسعه راه‌حل‌ها استفاده می‌کند. از آنجایی که ماژول‌ها کوچک‌ترین عنصر در یک برنامه هستند، اجرای UTP ها می‌تواند به تیم‌ها کمک کند تا بررسی کنند که هر قسمت از برنامه به درستی کار می‌کند. همچنین می‌تواند به آن‌ها کمک کند تا علت اصلی مشکلات احتمالی را پیدا کنند و بتوانند به سرعت آن‌ها را حل کنند.

## تست یکپارچه‌سازی (Integration testing)

در عین حال که تیم مشخص می‌کند چه اجزایی باید پیاده‌سازی شوند، می‌تواند پروتکل‌های تست یکپارچه‌سازی را هم در نظر داشته باشد و پیش برد. در مرحله آزمایش یکپارچه‌سازی، تیم‌های توسعه نقشه‌های تست یکپارچه‌سازی را که در مرحله طراحی معماری ایجاد کرده‌اند، دوباره بررسی می‌کنند.

آن‌ها از این تست‌ها برای ارزیابی اینکه آیا واحدهای منفرد درون برنامه می‌توانند در کنار یکدیگر وجود داشته باشند و در صورت لزوم با یکدیگر ارتباط برقرار کنند، استفاده می‌کنند. هنگامی که تیم طرح‌های تست یکپارچه‌سازی را اجرا می‌کند، مدیر پروژه گزارشی را برای به اشتراک گذاشتن نتایج آزمایش با مشتری و سایر ذی‌نفعان کلیدی تهیه می‌کند.

## تست سیستم (System testing)

در مرحله تست سیستم، تیم تست‌های سیستمی را که در مرحله طراحی سیستم توسعه داده‌اند، بررسی می‌کند. این تست‌ها بر روی اطمینان از اینکه پروژه تکمیل شده تمام نیازهای مشتری را برآورده می‌کند تمرکز دارد. تیم به جای آزمایش واحدها و اجزای جداگانه، کل سیستم را در این مرحله آزمایش می‌کند. این کار به آن‌ها درک جامعی از عملکرد کلی، وابستگی متقابل و عملکرد ارتباطی برنامه ارائه می‌دهد. برخی از انواع تست‌های سیستمی که ممکن است در این مرحله انجام شوند عبارتند از تست استرس، تست بار و تست رگرسیون.

## تست پذیرش (Acceptance testing)

در مرحله تست پذیرش، تیم از آزمون‌های پذیرش کاربر (UATs) ایجاد شده در مرحله تجزیه و تحلیل نیازمندی‌های کسب‌وکار استفاده می‌کند تا تعیین کند آیا برنامه با نیازهای کاربر مطابقت دارد یا خیر. تیم‌ها معمولاً UAT ها را برای جمع‌آوری داده‌های دقیق با کاربرهای واقعی انجام می‌دهند. این کار به آن‌ها کمک می‌کند تا قبل از تکمیل پروژه، زمینه‌های قابل بهبود را شناسایی کنند. اگر برنامه تمام UAT ها را پاس کند، به این معنی است که محصول برای استفاده افراد آماده است.