



## فصل سوم

# بررسی نگهداری و پشتیبانی سیستم

### هدف‌های رفتاری

اصول نگهداری و پشتیبانی سیستم را بشناسد.

## ۱- مقدمه

بعد از اینکه سیستم نرم افزاری تولید شد و توانست نیازهای کاربرانش را تأمین کند، هنوز کار یک پروژه نرم افزاری به اتمام نرسیده است، زیرا تازه نوبت نگهداری از آن نرم افزار می‌رسد و هر لحظه ممکن است تغییراتی در نرم افزار از سوی کاربر درخواست شود یا حتی نرم افزار با مشکلی غیرقابل پیش‌بینی رو به رو شود که نیاز به نگهداری نرم افزار را آشکار می‌سازد. رویه نگهداری نرم افزار بسیار شبیه و به سختی رویه تولید نرم افزار است و بر مبانی زیر تأکید می‌کند:

- کنترل رویه‌های سیستم

- کنترل روی تغییرات سیستم

- ایجاد تغییرات برای افزایش کارایی سیستم

- جلوگیری از کاهش کیفیت و کارایی سیستم

مهم‌ترین تغییرات از نیاز به تطبیق نرم افزار برای تعامل با موجودیت‌های خارجی مانند افراد، سازمان‌ها و سیستم‌های مصنوعی ناشی می‌گردد. در واقع به دلیل اینکه نرم افزار خیلی انعطاف پذیر است، اغلب به عنوان آسان‌ترین قسمت برای تغییرات در سیستم دیده می‌شود.

## ۲- نگهداری و پشتیبانی از نرم افزار

نگهداری و بهبود نرم افزار به منظور ایجاد توانایی مواجهه با مشکلات تازه کشف شده و نیازهای جدید می‌تواند زمانی بسیار بیشتر از تولید اولیه نرم افزار را صرف خود کند. در این مرحله نه تنها این امکان وجود دارد که مهندس نرم افزار مجبور به اضافه کردن کد به بخش‌هایی از نرم افزار اولیه گردد، که ممکن است در چهارچوب اولیه نگنجد، بلکه حتی درک اینکه نرم افزار اولیه چگونه کار می‌کند، می‌تواند نیاز به کوشش فراوانی از سوی مهندس نرم افزار داشته باشد. حدود دو سوم کار مهندسی نرم افزار نگهداری کد موجود می‌باشد. البته این آمار می‌تواند گمراه کننده باشد چرا که تنها بخش کوچکی از این زمان و انرژی صرف اشکال‌زدایی (دیاگ) می‌گردد. در حقیقت بیشتر این زمان و انرژی صرف گسترش سیستم برای انجام دادن وظایف جدید، که در واقع می‌توان آن را کار(پروژه/سفرانش) جدید محسوب

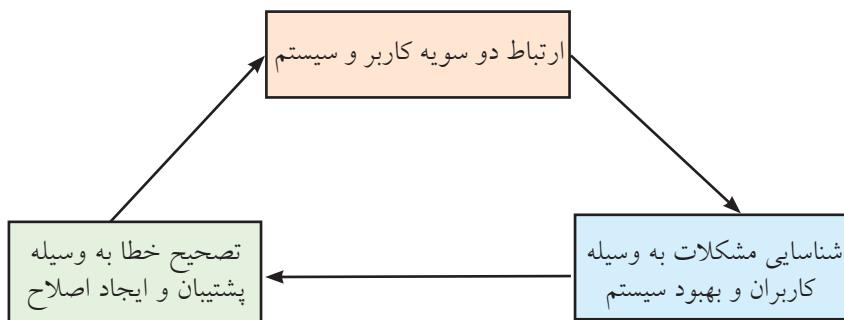
کرد، می‌شود. نکته قابل توجه این است که دو سوم انرژی و زمان مهندسان عمران و معماران و پیمانکاران ساختمانی نیز به نحو مشابهی صرف تغییرات می‌شود.

## تعريف تعمیر و نگهداری

تعمیر نرم افزار فعالیت خیلی بزرگی است که اغلب به عنوان تمام اعمالی که روی سیستم نرم افزاری بعد از آنکه سیستم قابل استفاده شد، انجام می‌گیرد تعریف می‌گردد. این فعالیت تمام اعمال اصلاح خطاهای افزایش، حذف و اضافه کردن قابلیت‌های جدید، انطباق با تغییرات در داده‌های مورد نیاز و محیط‌های عمل، بهبود کارایی، قابلیت استفاده یا هر صفت کیفی دیگر را پوشش می‌دهد.

## تعريف تعمیر نرم افزار از دید IEEE

تعمیر نرم افزار عبارت است از فرایند اصلاح سیستم نرم افزاری یا اجزای آن بعد از تحویل سیستم، برای رفع خطاهای، بهبود کارایی، تطبیق با تغییرات محیط یا دیگر خصوصیات.



این تعریف یک دید معمولی از تعمیر نرم افزار، که یک فعالیت بعد از تحویل است را معرفی می‌کند. تعمیر نرم افزار هنگامی که یک سیستم به مشتری یا کاربر تحویل داده می‌شود، شروع می‌گردد و همه فعالیت‌هایی که از عملیات سیستمی نگهداری می‌کند و با نیازهای کاربر مواجه است را در بر می‌گیرد. تعمیر نرم افزار را می‌توان به عنوان یک بخش اصلی از دوره حیات سیستم برای مدیریت تغییرات سیستم نرم افزاری پذیرفت.

تعمیر نرم افزار مجموع فعالیت‌هایی است که برای فراهم کردن هزینه مؤثر پشتیبانی از یک سیستم نرم افزاری مورد نیاز است. فعالیت‌ها در طول مراحل قبل از تحویل و بعد از تحویل اجرا می‌گردند. فعالیت‌های قبل از تحویل سیستم شامل برنامه‌ریزی برای عملیات بعد از تحویل سیستم است که قابل پشتیبانی و تصمیم‌گیری است. فعالیت بعد از تحویل شامل اصلاح نرم افزار، آموزش، یک عامل کمکی و راهنمایی باشد.

## دسته‌بندی‌های تعمیر و نگهداری نرم افزار

دسته‌بندی‌ها به فهم بهتر اهمیت تعمیر و تأثیر آن روی هزینه و کیفیت سیستم برای استفاده کمک می‌کنند.

### ۱- نگهداری اصلاحی (Corrective Maintenance)

تغییراتی که بر روی تولیدات نرم افزاری بعد از تحویل سیستم برای رفع خطاهای کشف شده انجام می‌گیرد شامل همه تغییراتی است که موجب حذف فعالیت‌های معیوب در نرم افزار می‌گردد.

### ۲- نگهداری تطبیقی (Adaptive Maintenance)

تغییراتی که روی تولیدات نرم افزاری بعد از تحویل، برای حفظ قابل استفاده بودن برنامه در محیط‌های تغییر یافته یا در حال تغییر صورت می‌گیرد. شامل تغییرات مورد نیاز برای تطبیق سیستم با محیطی که سیستم باید در آن فعالیت داشته باشد. برای مثال تغییرات یک سیستم برای کار آن بر روی پلتفرم سخت افزار جدید، سیستم عامل جدید.

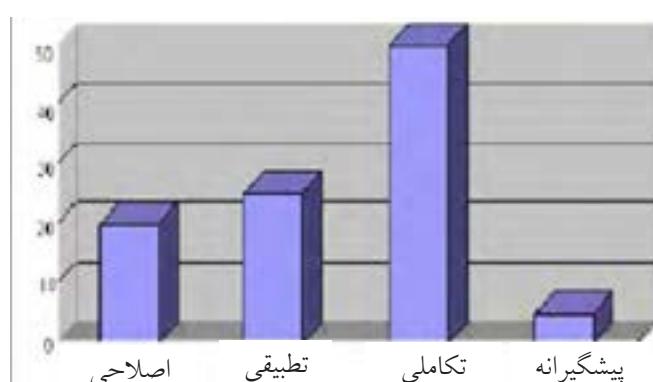
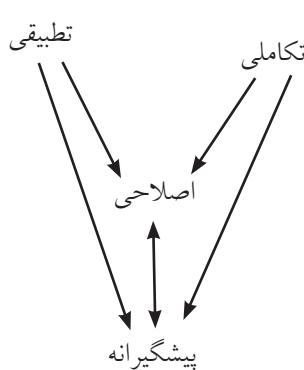
### ۳- نگهداری تکاملی (Perfective Maintenance)

تغییراتی که روی تولیدات نرم افزاری بعد از تحویل برای بهبود کارایی یا قابل تعمیر بودن صورت می‌گیرد. شامل تغییراتی است که از جانب درخواست‌های کاربر سازماندهی می‌شود. برای مثال می‌توان به مواردی چون اضافه کردن، حذف و اصلاح توابع، بازنویسی مستندات، بهبود کارایی یا بهبود سهولت استفاده اشاره کرد. حالت ایده‌آل این است که عملیات تعمیر نباید قابلیت اطمینان و قابلیت تعمیر را کاهش دهد یا ساختار اجزای سیستم را به هم زند. در غیراین صورت تغییرات آینده به تدریج مشکل‌تر و فاصله هزینه پیاده سازی آنها بیشتر و گران‌تر خواهد شد.

### ۴- نگهداری پیشگیرانه (Preventive Maintenance)

شامل همه اصلاحاتی است که سبب می‌شود یک قطعه نرم افزار قابل تعمیر شود.

نگهداری اصلاحی زمان بندی نشده Emergency Maintenance که برای حفظ قابل استفاده بودن سیستم اجرا می‌شود.



## رابطه بین انواع دسته بندی ها

هزینه نگهداری سیستم های نرم افزاری تقریباً معادل نیمی از هزینه ساخت تمامی مراحل نرم افزار می شود. البته، نباید این نکته را نادیده گرفت که این هزینه کاملاً به نوع نرم افزار، نوع فناوری های استفاده شده در نرم افزار، نوع طراحی، مستندات و کیفیت روش (COCOMO) آزمایشی که روی نرم افزار انجام شده، بستگی دارد که می توان از روش اندازه گیری هزینه تولید نرم افزار نیز برای اندازه گیری این معیار استفاده کرد.

## ابزارهای نگهداری

تکنیک هایی که می توان در نگهداری نرم افزارها از آن استفاده کرد، عبارتند از:

- مدیریت تغییرات، نگهداری و رد گیری تمامی تغییراتی که در سیستم به وجود می آید به همراه تاریخ، ساعت و نوع تغییر و توضیح در مورد اشکال یا تغییرات اصلاحی.
- مدیریت ساختار و حفظ ساختار اصلی برنامه بدون این که در عملکرد آن مشکلی به وجود آید.
- آنالیز تأثیرگذاری (برای محاسبه هزینه نگهداری).
- ابزارهای خودکار برای شناسایی تفاوت دو فایل برنامه.
- ابزارهای شناسایی اشکال و Debugging.
- ابزارهای نوشتگر برنامه که به صورت خودکار تغییرات را نیز رکورد می کنند و در زمان مورد نیاز آن تغییرات را اعمال می کنند.
- ابزارهای اندازه گیری کدها (برای اندازه گیری هزینه نگهداری نرم افزار براساس تعداد خط کد و پارامترهای دیگر برنامه نویسی مانند درجه سختی زبان برنامه نویسی).

## مشکلات تعمیر نرم افزار

بزرگ ترین مشکلات تعمیر نرم افزار عبارتند از: درک برنامه، آنالیز ضربه ای و تست برگشت.

۱- درک برنامه (Program Comprehension)

هنگامی که یک تغییر روی قطعات نرم افزاری اتفاق می افتد این مسئله خیلی مهم است که نگهدارنده سیستم درک

کاملی از ساختار، رفتار و توابع سیستم در شروع عملیات اصلاح به دست بیاورد. یکی از اساسی ترین این شناخت‌ها می‌تواند تولید یک طرح پیشنهادی اصلاح برای انجام هدف تعمیر باشد. نگهدارندگان سیستم مقدار زیادی از وقت‌شان را صرف خواندن کد و مستندات زمینه برای درک منطق، هدف و ساختار سیستم می‌کنند. برآوردهای موجود نشان می‌دهد زمانی که برای درک برنامه برای عمل تعمیر صرف می‌شود بین ۰۵ تا ۰۹ درصد متغیر است و نگهدارنده سیستم خیلی به ندرت پیش می‌آید که نویسنده کد برنامه باشد و مستندات کامل و به روز به ندرت در دسترس است.

## ۲- آنالیز ضربه‌ای (Impact Analysis)

یکی از چالش‌های مهم در تعمیر نرم افزار تصمیم گیری برای تأثیر اصلاحات پیشنهادی روی سیستم موجود است. آنالیز ضربه‌ای شامل عمل برآورد برای تغییر با هدف کمینه کردن تأثیرات غیرمنتظره می‌باشد. وظیفه آنالیز ضربه‌ای برآورد تغییرات پیشنهادی، ارزیابی ریسک پذیری پیاده سازی و برآورد تأثیر آن روی منابع، تلاش و زمانبندی می‌باشد.

## ۳- تست برگشت (Regression Testing)

بعد از اینکه یک تغییر روی سیستم پیاده شد، سیستم نرم افزاری برای اطمینان از اینکه اجرای آن بر طبق اصلاحات امکان پذیر است دوباره تست می‌شود. فرایند تست سیستم بعد از اصلاح آن تست برگشت نامیده می‌شود که به دو منظور انجام می‌شود.

۱ - کسب اطمینان از اینکه تغییرات درست است.

۲ - اطمینان از اینکه قسمت‌های تغییر نیافته سیستم تحت تأثیر قرار نگرفته‌اند.

تست برگشت از تستی که در زمان توسعه سیستم اجرا می‌گردد متفاوت است. بدین معنی که یک مجموعه از موارد تست ممکن است برای استفاده مجدد در دسترس باشد. تغییراتی که در طول فرایند تعمیر به وجود می‌آید، معمولاً کوچک است و بنابراین اجرای معمولی همه موارد تست بعد از هر تغییر ممکن است خوبی هزینه‌بر باشد. هزینه ردیابی خطاهای در مراحل بعدی در مدل چرخه حیات افزایش می‌یابد.

# ۳-۵

## مهندسی معکوس (Reverse Engineering)

فرایند تجزیه و تحلیل سیستم برای شناسایی اجزا و ارتباطات و روابط آنها با هم و ایجاد نمایش سیستم در دیگر اشکال یا در یک سطح انتزاعی بالاتر می‌باشد. مهندسی معکوس فرایندی نیست که سیستم را تغییر دهد بنابراین درگیر تغییر

سیستم نرم افزاری در هنگام آزمایش نمی‌گردد.

برخی از اهداف مهندسی معکوس: تهیه یک نسخه کپی سیستم به همراه پیچیدگی‌های آن، تولید دیدهای متناوب، پوشاندن اطلاعات گم شده، شناخت تأثیرات، ترکیب انتزاع‌های سطوح بالاتر، آسان‌سازی استفاده مجدد و ... می‌باشد.

برخی از حوزه‌های کاربرد مهندسی معکوس عبارتند از: شناسایی قبل استفاده مجدد بودن دارایی‌ها، پیدا کردن اشیاء در برنامه‌های رویه‌ای، کشف معماری، محرك ادراکی مدل‌های داده، شناسایی نسخه برداری‌ها و تکرارها، انتقال برنامه‌های دودویی در کد منبع، تجدید کردن رابط‌های کاربردی، برنامه‌های ترتیب موازی، ترجمه، فرایند انتقال برنامه از سیستم اصلی به سیستم‌های کوچک‌تر و ارزان‌تر، مهاجرت و بسته بندی کدهای موروثی.

## سازماندهی مجدد

سازماندهی مجدد عبارت است از احیاء نظم آغازین یک سیستم. سازماندهی مجدد در موقعی انجام می‌شود که نظم آغازین سیستم دچار اختلال شده و سیستم کارایی خود را از دست می‌دهد. با احیاء نظم آغازین سیستم و انجام عملیات سازماندهی مجدد، کارایی سیستم به حالت اولیه بر می‌گردد.

## مهندسی مجدد (Re-Engineering)

مهندسی مجدد بر روی یک سیستم نرم افزاری به منظور درک بهتر و نگهداری طولانی‌تر آن صورت می‌گیرد. مهندسی مجدد نرم افزار هر فعالیتی است که باعث درک نرم افزار گردد یا باعث فراهم کردن یا بهبود خود نرم افزار شود که معمولاً برای افزایش قابلیت تعمیر، قابلیت استفاده مجدد یا قابلیت استنتاج کردن می‌باشد. (آزمایش و دگرگونی یک سیستم برای به هم پیوستن و پیاده سازی آن در فرم جدید).

– مهندسی مجدد می‌تواند به کاهش ریسک تکامل یک سازمان و به دست آوردن مبلغ سرمایه گذاری شده در نرم افزار کمک کند.  
– می‌تواند به آسان‌تر شدن نرم افزار برای تغییر کمک کند.

– مهندسی مجدد یک تجارت بزرگ است.

– CASE قابلیت توسعه ابزار را دارد.

– یک عامل برای خودکار کردن تعمیر نرم افزار به شمار می‌رود و به عنوان یک سازمان دهنده برای به کار بستن تکنیک‌های هوش مصنوعی برای حل مشکلات مهندسی مجدد نرم افزار می‌باشد.

## خلاصه فصل

رویه نگهداری نرم افزار بسیار شبیه و به سختی رویه تولید نرم افزار است و بر مبانی زیر تأکید می کند: کنترل روی رویه های سیستم، کنترل روی تغییرات سیستم، ایجاد تغییرات برای افزایش کارایی سیستم، جلوگیری از کاهش کیفیت و کارایی سیستم.

نگهداری و بهبود نرم افزار به منظور ایجاد توانایی مواجهه با مشکلات تازه کشف شده و نیازهای جدید می تواند زمانی بسیار بیشتر از تولید اولیه نرم افزار را صرف خود کند. در این مرحله نه تنها این امکان وجود دارد که مهندس نرم افزار مجبور به اضافه کردن کد به بخش هایی از نرم افزار اولیه گردد که ممکن است در چهار چوب اولیه نگنجد، بلکه حتی درک اینکه نرم افزار اولیه چگونه کار می کند می تواند نیاز به کوشش فراوانی از سوی مهندس نرم افزار داشته باشد.

تعمیر نرم افزار مجموعه فعالیت هایی است که برای فراهم کردن هزینه مؤثر پشتیبانی از یک سیستم نرم افزاری مورد نیاز است. فعالیت ها در طول مراحل قبل از تحویل و بعد از تحویل اجرا می گردند. فعالیت های قبل از تحویل سیستم شامل برنامه ریزی برای عملیات بعد از تحویل سیستم است که قابل پشتیبانی و تصمیم گیری است. فعالیت بعد از تحویل شامل اصلاح نرم افزار، آموزش، یک عامل کمکی و راهنمایی باشد.

بزرگ ترین مشکلات تعمیر نرم افزار عبارتند از: درک برنامه، آنالیز ضربه ای و تست برگشت.

## خود آزمایی

۱- به صورت یک فعالیت عملی و پژوهشی، ابتدا پژوهه سیستم پرداخت حقوق کارمندان دولت قبل از لایحه مدیریت خدمات کشوری را نوشه و آن را مستندسازی نمایید. پژوهه را حداقل برای ۵۰ کارمند نمونه پیاده سازی کنید. در بخش دوم، همان پژوهه را با لایحه خدمات کشوری که بر مبنای امتیاز می باشد، تبدیل کنید و مستند سازی آن را انجام دهید.

۲- مدل های تعمیر و نگهداری نرم افزار را به اختصار توضیح دهید.

۳- تحقیق کوتاهی در مورد مهندسی معکوس انجام داده و آن را به کلاس ارائه دهید.