

مراحل بازمهندسی اتوماسیون اداری

بازگویی دلایل ضروری برای بازمهندسی نرم افزارها به ویژه در زمینه نرم افزارهای اتوماسیون اداری را در قسمت اول یعنی دلایل و اهمیت [بازمهندسی اتوماسیون اداری](#) مقاله به طور کامل توضیح دادیم. در بخش دوم این مقاله تلاش می کنیم تا از جزئیات و مراحل بازمهندسی نرم افزارها سخن بگوییم و فرایندهای پنج گانه آن را به اختصار تعریف و بازگویی کنیم.

حقیقت این است که برای بازمهندسی نرم افزارها فرایندهای مختلفی ایجاد شده است که طی آن سعی می شود کلیه عملیات های بازمهندسی در چارچوب مشخصی صورت بگیرند. این فرایندها ممکن است در برخی از فازهای با هم متفاوت باشند؛ اما معمولاً تمامی آنها از یکسری اصول اولیه، تبعیت می کنند. در اینجا مجموعه ای از فازهای تقریباً مشترک در بین تمامی این فرایندها که بر مبنای اصول بازمهندسی آورده شده اند را بیان می کنیم. در مجموع می توان گفت که فرایند بازمهندسی نرم افزار شامل فازهای اصلی زیر است:

- مهندسی معکوس
- بازطراحی
- بازسازی کد
- تست
- استقرار و بازخوردگیری

مهندسی معکوس

مهندسی معکوس از فازهای اصلی مربوط بازمهندسی با هر هدفی از اجرای این عملیات است. به صورت خیلی ساده می‌توان گفت مهندسی معکوس عملیاتی است که طی آن از روی منابع کد نرم‌افزار به طراحی، قوانین و مقررات کسب و کار نرم‌افزار که همان نیازمندی‌های اولیه است، می‌توان دست پیدا کرد. در مهندسی نرم‌افزار، نیازهای نرم‌افزاری تحلیل، طراحی و سپس به کد تبدیل می‌شوند. مستندات مختلفی از نرم‌افزارها در فازهای مختلف در زمان تولید وجود می‌آیند که به مرور زمان، قدیمی و ناکارآمد می‌شوند.

آنها بر اساس تغییراتی که در طول زندگی خود داشته‌اند از درجه اعتبار ساقط می‌شوند؛ بنابراین نیاز است به هر هدفی که عملیات بازمهندسی می‌خواهد صورت بگیرد ابتدا قواعد، قوانین و رفتارهای فعلی نرم‌افزار به صورت جامع شناسایی شود. همه این اقدامات در فاز مهندسی معکوس، امکان‌پذیرند. در این فاز، منبع اصلی استخراج اطلاعات، کدهای مربوط به نرم‌افزار است. افراد درگیر در پروژه بازمهندسی، ابتدا کدهای نرم‌افزار را می‌خوانند و این کدها را به نمودارهای طراحی، معماری و سپس به داستان‌های کاربری که شامل رفتارهای سیستم، قواعد و قوانین کسب و کار است، تبدیل می‌کنند.

مهم است که مشخص شود رفتار سیستم در حال حاضر چگونه است. این کار با فرایندی پایین به بالا سعی می‌کند از روی کد به انتزاع لازم در بالاترین سطح طراحی رسیده و خروجی‌های لازم را ایجاد کند. خروجی این فاز، مجموعه‌ای از مستندات

طراحی و تحلیل را شامل می‌شود که در ادامه فرایند پروژه بازمهندسی، مشخص‌کننده دقیق تغییرات خواهند بود. این فاز در نرم‌افزارهای بزرگ بسیار پر هزینه است و افرادی که این کار را انجام می‌دهند باید پیش از شروع کار، آموزش‌های لازم برای تبدیل کدهای نرم‌افزار به داستان‌های کاربری دقیق را گذرانده باشند.

باز طراحی

فاز بازطراحی معمولاً در زمان‌هایی اجرا می‌شود که هدف از بازمهندسی، تغییر رفتار و یا تغییر معماری نرم‌افزار باشد. در این فاز در صورتیکه هدف، تغییر معماری باشد عملیات‌های بازطراحی، مشخص‌کننده ساختار نرم‌افزار بعد از اعمال تغییرات خواهند بود و در صورتیکه هدف، تغییر رفتارهای نرم‌افزار باشد عملیات‌های بازطراحی، رفتار و معماری نرم‌افزار بر اساس رفتارهای جدید را رقم می‌زند.

این کار بر اساس عملیاتی بالا به پایین سعی می‌کند دوباره و بر اساس هدف مشخص شده، طراحی مجددی را صورت دهد. برای این کار از بالاترین سطح انتزاع به دست آمده از عملیات مهندسی معکوس استفاده می‌شود و برای هر سطح مجموعه، طراحی‌هایی صورت می‌گیرد که بر اساس هدف مشخص شده در برگزیده نتیجه نهایی نرم‌افزار است.

مستندات تولید شده در این فاز، مشابه مستندات مهندسی معکوس هستند که بر اساس هر کدام از آنها نقشه راهی برای تبدیل نرم‌افزار فعلی به نرم‌افزار نهایی به دست می‌آید. در این فاز در بالاترین سطح انتزاع ابتدا مشخص می‌شود هر رفتار نرم‌افزار پیش از بازمهندسی به چه رفتاری تبدیل خواهد شد. سپس آرام آرام به سطح پایین‌تر آمده و

معلوم می‌شود که در طراحی فعلی، نرم‌افزار به چه طراحی‌ای تبدیل می‌شود. در این ساختار هر چقدر رفتارهای اولیه نرم‌افزار بیشتر دستخوش تغییرات شوند نتیجه بازطراحی فاصله بیشتری از وضعیت فعلی خواهند داشت و در نهایت، ماژول‌های تولیدی بیشتر تغییر می‌کنند.

بر اساس بزرگی میزان تغییرات مورد نیاز در این فاز گاهی برخی از ماژول‌های نرم‌افزار حذف و ماژول‌های جدید تولید می‌شوند. گاهی ماژول‌ها باقی می‌مانند و برخی از رفتارهای آنها منقضی و رفتارهای جدید جایگزین آن می‌شوند و گاهی همان رفتارها به معماری جدید و ساختار تکنولوژیک متفاوتی، منتقل می‌شوند. همچنین در بازطراحی ممکن است ساختار داده‌ای نرم‌افزار مورد بازبینی و بر اساس مهندسی معکوس انجام شود. می‌توان با بازتعریف اشیا و مدل‌های داده‌ای موجود در نرم‌افزار مشکلات پیشین در ساختار داده‌ای را رفع و ساختار داده‌ای جدیدی ایجاد کرد که بتواند رفتارهای نرم‌افزار را بهبود ببخشد. در انتهای این فاز، نقشه راهی برای تبدیل کدهای فعلی به کدهای مطلوب به دست می‌آید.

بازسازی کد

این فاز از بازمهندسی نیز از مراحل اصلی برای هر هدفی از این عملیات، است. در این فاز، کدهای جدید پیاده‌سازی می‌شوند. اگر هدف از بازمهندسی تغییر معماری یا تغییر رفتار باشد بر اساس نقشه راه به دست آمده از فاز بازطراحی، کدهای جدید برای تغییر رفتار و یا معماری نوشته و تست می‌شوند. اگر هدف ریفکتور کد باشد، کدها بدون تغییر در رفتار و معماری طوری تغییر می‌کنند که کد نهایی قابل پشتیبانی و توسعه‌پذیر باشد.

عملیات بازسازی کد شاید پیچیده‌ترین فاز بازمهندسی نرم‌افزار باشد؛ زیرا همواره تغییر دادن کد موجود به کدی جدید که در کنار تغییرات احتمالی رفتاری و معماری، قابلیت پشتیبانی و توسعه داده باشد و مشکلات پیش از بازمهندسی را نداشته باشد بسیار سخت‌تر از نوشتن یک نرم‌افزار جدید است. در طول حیات چند ساله یک نرم‌افزار، چسبندگی‌های بین ماژول‌ها و قسمت‌های مختلف یک ماژول بهم به مرور زمان بیشتر و بیشتر می‌شود؛ هر چه میزان این وابستگی‌ها بیشتر باشد تغییر دادن هر قطعه کد، تاثیرات بسیاری در سایر قسمت‌های نرم‌افزار ایجاد می‌کند. مدیریت این تغییرات در طول بازسازی کد از عوامل پیچیده‌شدن این فاز در بازمهندسی است. در انتهای این فاز اطمینان حاصل می‌شود که کدهای بازسازی شده هیچیک از ناهنجاری‌های پیشین را نداشته باشند.

تست‌گیری

در این فاز، نرم‌افزار به صورت جامع تست می‌شود تا پیش از انتشار، مجموعه باگ‌ها و مشکلات آن پیدا و رفع شوند. تست‌هایی مانند تست واحد، یکپارچگی، تست بار، کارآیی و امنیت مجموعه تست‌هایی هستند که بر اساس اهمیت بخش‌های مختلف نرم‌افزار و بودجه تعیین شده برای بازمهندسی صورت می‌گیرند. نتایج این تست‌ها در بهبود کیفیت نرم‌افزار تولید شده بعد از بازمهندسی، تعیین کننده‌اند.

استقرار و پشتیبانی

فاز آخر از بازمهندسی به استقرار و پشتیبانی از نرم‌افزار اختصاص دارد. در مرحله،

لازم است تا مستندات راهنما به روز شوند، آموزش‌های لازم برای استفاده از نرم‌افزار (در صورت تغییر در رفتار نرم‌افزار) به برای مشتریان داده شود و عملیات‌های پشتیبانی برای رفع عیوب و بازخوردگیری از نرم‌افزار به صورت کامل انجام شوند.

نتیجه‌گیری

در این مقاله مجموعه‌ای از مفاهیم اصلی و بنیادی مربوط به بازمهندسی نرم‌افزارها را با هدف بیان چرایی و چگونگی آن و همچنین خلاصه‌ای از فرآیندهای عملیاتی آن را بیان کردیم. نکته مهم این است که همه این مفاهیم، دانش و اصولی هستند که به صورت تئوری بیان می‌شوند. در عمل برای آنکه بتوانیم عملیات بازمهندسی را انجام دهیم نیاز است که کمی جامع‌تر و عملگراتر به این مقوله نگاه شود.

باید بتوانیم با استفاده از این مفاهیم و تعریف صحیح اهداف، فرایندهای مربوط به هر کدام از فازهای مختلف بازمهندسی نرم‌افزار را بر اساس مشخصه‌هایی مانند توان منابع انسانی، هزینه‌ها، چالش‌های آموزشی، مشکلات نرم‌افزاری و ارتقای سطح فرهنگی مشتریان تعریف و اجرا کنیم.

در قسمت سوم این مقاله از تجربیات واقعی پیاده‌سازی شده برای بازمهندسی نرم‌افزارهای حوزه [اتوماسیون اداری دیدگاه](#) سخن می‌گوییم. تلاش می‌کنیم تا درباره مشکلات و راه‌حل رفع آن‌ها از طریق بازمهندسی، فرایند پیاده‌سازی شده برای بازمهندسی نرم‌افزارها به ویژه در حوزه اتوماسیون اداری و همچنین ف مدیریت کنترل کیفیت نرم‌افزارها سخن بگوییم و تلاش کنیم تا تجربه واقعی بازمهندسی نرم‌افزارهای

اتوماسیون اداری را بر اساس هر سه هدف بیان شده با شما در میان بگذاریم.

پایان قسمت دوم

قسمت سوم: چالش‌های نرم‌افزارهای اتوماسیون اداری

چارگون