

مقایسه مهندسی مجدد و معکوس و پیشرو در مهندسی نرم افزار و بررسی یک راهکار پیشرفته و کارا برای مهندسی مجدد

سارا سکوت^۱، تقی جاودانی گندمانی^{۲*}

۱- دانشجوی کارشناسی ارشد نرم افزار، دانشگاه آزاد اسلامی، واحد اصفهان (خوراسگان)، اصفهان، ایران،
sarasokoot@gmail.com

۲- استادیار مهندسی نرم افزار، دانشگاه آزاد اسلامی، واحد اصفهان (خوراسگان)، اصفهان، ایران،
javdani@ieeee.org

چکیده

امروزه موسسات بزرگ نه تنها با مشکل جایگزینی سیستم های اطلاعاتی قبلی با سیستم های مشابه جدید مواجه می باشند ، بلکه آنها مجبور به ذخیره و نگهداری برنامه ها و داده های قدیمی خود هستند . مهندسی معکوس در راستای بدست آوردن نمایش سطح انتزاعی از برنامه ها ، به کسب دانش از دست رفته و ایجاد مجدد سیستم پیچیده می پردازد که انتقال سیستم قدیمی به سیستم جدید و نگهداری ساختار آن از جمله کارهای مهندسی معکوس است . مهندسی مجدد پس از تشخیص مشکلات نرم افزار قدیمی سبب بهبود کیفی آن و سازگارمندی بیشتر با چهارچوب ها می گردد. این مقاله ابتدا به تعریف مفاهیم پایه ای مهندسی معکوس و مهندسی مجدد و مهندسی پیشرو پرداخته و سپس به تفاوت بین آنها و در ادامه به تفصیل به مهندسی مجدد و در نهایت یک روش کارا و پیشرفته را در این زمینه بررسی نموده است.

کلمات کلیدی: مهندسی معکوس، مهندسی مجدد، مهندسی پیشرو و مهندسی مجدد کارا.

۱- مقدمه

امروزه جوامع در حال تجربه کردن نوع جدیدی از اقتصاد به عنوان اقتصاد دانش محور هستند که در آن دانش و تکنولوژی به عنوان کلید اصلی ایجاد کننده مزیت رقابتی پایدار، نوآوری و تولید ثروت شناخته می شود. بر این اساس ضرورت مدیریت تکنولوژی بیش از پیش نمود می یابد.

کشورهای پیشرفته توجه ویژه ای به امر خلق و توسعه تکنولوژی دارند و در سطوح مختلف مدیریت تکنولوژی برای آن برنامه ریزی می کنند اما در کشورهای درحال توسعه به دلیل عقب ماندگی زیاد این کشورها و همچنین به دلیل عدم توان درخلق تکنولوژی، پیمودن راه انتقال تکنولوژی با هدف پرکردن شکاف تکنولوژی گریزناپذیر است و برای موفقیت هرچه بیشتر در این زمینه لازم است که علاوه برآماده سازی زیرساخت های موجود، فرآیند انتقال تکنولوژی را با تمرکز بیشتری مدیریت کنند [۱].

یکی از راه های دستیابی به تکنولوژی انتقال تکنولوژی است که درآن با استفاده از منابع خارجی و خرید آن از خارج از بنگاه به تکنولوژی مورد نظر دسترسی پیدا می شود و یکی از راههای انتقال تکنولوژی استفاده از مهندسی معکوس و مهندسی مجدد می باشد. از این رو در این مقاله ابتدا هر یک را جداگانه بررسی موشکافانه خواهد نمود و در نهایت به مقایسه این دو خواهد پرداخت.

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

عبارت «مهندسی معکوس» از مفاهیم سخت افزاری گرفته شده است که طی آن یک سازمان سخت افزاری، سخت افزار تولید شده توسط شرکت رقیب را خریداری می کند تا با تجزیه آن به رازهای ساخت آن دست پیدا می کند. این رازها هنگامی که طراحی رقیب و خصوصیات ساخت آن به دست آید، برملا می شود. اما این مستندات برای شرکت های رقیب قابل دسترسی نیستند. مهندسی معکوس به وسیله آزمایش خصوصیات یک محصول، یک یا چند خاصیت طراحی آن را به دست می آورد. مهندسی مجدد و الگوهای مهندسی مجدد از مفاهیم نسبتاً جدیدی است که در ابتدا برای تاثیر گذاشتن بر جامعه ی مهندسی نرم افزار شروع شد. با تغییر منابع با هدف تجدید سازمان از سیستم های قدیمی به سوی تمرکز بر پیشرفت نرم افزار جدید، تجارت قادر خواهد بود زمان و منابع با ارزش زیادی را ذخیره نماید. اگرچه مزایای این رویکرد به وضوح دیده می شود اما برخی مشکلات از جمله بی تجربگی و کمبود ابزار مناسب، مهندسی مجدد سیستم را کمی مشکل می کند [۲].

۲- مفاهیم اصلی

در ابتدا به تعریف مفاهیم اصلی در زمینه مهندس معکوس و مهندسی مجدد پرداخته می شود:

2-1- تعاریف مختلف مهندسی معکوس

تاکنون تعاریف مختلفی از مهندسی معکوس ارائه شده است:

دریک تعریف، مهندسی معکوس فرآیندی آگاهانه و عالمانه در دسترسی به تکنولوژی موجود است [۳]. از دیدگاه دیگر، مهندسی معکوس را می توان فرآیندی از توسعه ی یک دسته از مشخصات برای یک سیستم سخت افزاری پیچیده به وسیله ی ارزیابی ها، آزمایش ها و پژوهش های منظم از روی نمونه های موجود آن سیستم دانست [۴]. در تعریف دیگر مهندسی معکوس عبارت است از فرآیند نسخه برداری از یک کالا از لحاظ عملکردی و ابعادی به وسیله ی آزمایش فیزیکی و اندازه گیری قطعات موجود برای توسعه ی اطلاعات فنی مورد نیاز [۵]. همچنین مهندسی معکوس را یک روش اصولی و معین برای تحلیل وسایل یا سیستم های موجود نیز می دانند [6]. مهندسی معکوس، یافتن فرآیند ترمیم طراحی است که ابزارهایش اطلاعات، معماری و ساختار زیر برنامه ای نرم افزار موجود را به دست می آورند.

2-2- تعاریف مهندسی مجدد

مهندسی مجدد برنامه، دوباره اجرا کردن سیستم های قدیمی برای افزایش قابلیت نگهداری است. مهندسی مجدد به صورت مستندسازی مجدد سیستم، سازماندهی، ساختارمند کردن مجدد سیستم، ترجمه ی سیستم به زبان برنامه نویسی مدرن، اصلاح کردن و به روز کردن ساختار است.

مهندسی مجدد توسط Chikofsky و Cross در مقاله ۱۹۹۰ آنها توصیف شده است "تست و تغییر سیستم برای بازتولید آن در شکل جدید". مهندسی مجدد، اصلاح یک سیستم نرم افزاری است که بعد از آن اتفاق می افتد، مهندسی معکوس، به طور کلی برای اضافه کردن قابلیت های جدید، و یا برای اصلاح اشتباهات صورت می گیرد این فرایند اغلب به اشتباه به عنوان مهندسی معکوس نامیده می شود. پس به طور کلی می توان گفت مهندسی معکوس بررسی اولیه سیستم است و مهندسی مجدد اصلاح بعدی است [7].

نگهداری از سیستم های قدیمی پرهزینه است. بنابراین مهندسی مجدد این سیستم ها عمر مفید آن ها را افزایش می دهد. مهندسی مجدد، ساختار سیستم را بهبود می بخشد، سیستم جدید را مستند و فهم آن را آسان می کند .

مهندسی مجدد، آزمون مدل و کاربرد سیستم قدیمی موجود را به یکدیگر ربط می دهد و تکنیک های مختلفی را برای طراحی مجدد سیستم به کار می گیرد. هدف از این کار، استفاده ی مناسب و بهتر از قبل نرم افزار است. البته این کار، فرآیند آسانی نیست زیرا به روز کردن و اضافه کردن عامل جدید ممکن است باعث عدم صحت و درستی و همین طور به روزآوری مستندات سیستم شود. به خصوص اگر سیستم به مردم تفویض نشود، مهارتی در راه تفکر مهندسی مجدد به وجود نمی آید که در نهایت سیستم توانایی ترمیم سریع و کارآیی خود را در صورت بروز خرابی از دست می دهد [8].

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

تلاش در جنبه های بنیادی مهندسی مجدد مانند نگهداری، مشتری مداری، کارکردی کردن و دیگر نیازمندی ها هر اندازه هم که کوچک باشد سودمند است. این موضوع الگوی مهمی را در مهندسی مجدد بیان می کند: "تغییر، اما به مقدار اندک ممکن" [9].

2-3- تعاریف مهندسی پیشرو

مهندسی پیشرو مخالف مهندسی معکوس است. در مهندسی پیشرو، مجموعه ای از اولویت های مورد علاقه، آنها را به سیستم کاری تبدیل می کند، و سپس آنچه را سیستم می تواند و نمی تواند انجام دهد مشاهده می کند. مهندسی پیشرو پایه و اساس روانشناسی مصنوعی است. مهندسی رو به جلو احتمال دارد نظریه های ساده تری را نسبت به مهندسی معکوس تولید کند، زیرا تمایل به ویژگی پیچیدگی های رفتاری در مکانیسم های داخلی عامل دارد. برای تئبرگ¹ این را قانون تجزیه و تحلیل بالارونده و ترکیب پایین رونده می نامد. یک سابقه طولانی مهندسی پیشرو نشان می دهد که عوامل ساده می توانند رفتارهای شگفت انگیز پیچیده ای ایجاد کنند در صورتی که در یک محیط جالب تعیبه شده باشند [10].

3- ویژگی های مهندسی مجدد

3-1- عوامل افزایش دهنده هزینه های مهندسی مجدد

- عوامل افزایش دهنده هزینه های مهندسی مجدد عبارتند از:
- کیفیت نرم افزار مهندسی مجدد شده: کیفیت پایین نرم افزار و مستندات مرتبط با آن.
 - ابزار در دسترس: اگر برای مکانیزه کردن بیشتر تغییرات برنامه از ابزار موردی و ویژه ی آن قسمت استفاده شود.
 - گسترش داده های تبدیلی مورد نیاز: اگر مهندسی مجدد نیاز به تبدیل و تغییر بخش اعظمی از داده ها داشته باشد.
 - دسترسی به کارمندان متخصص و خبره: اگر کارمند برای نگهداری سیستم مسئولیت پذیر نباشد.
- شایان ذکر است که مهم ترین زیان مهندسی مجدد نرم افزار محدودیت کاربردی آن است که از طریق مهندسی مجدد

3-2- فرآیندهای مهندسی مجدد

- در ادامه به بررسی فرآیندهای مهندسی معکوس و مهندسی مجدد پرداخته شده است [11]. همانطور که در شکل 1 نمایش داده شده، فعالیت های مربوط به این فرآیند عبارتند از:
- 1- ترجمه ی کد مبدا: برنامه از زبان برنامه نویسی قدیمی به نسخه های مدرن به همان زبان یا زبان های دیگر تبدیل می شود.
 - 2- مهندسی معکوس: برنامه تجزیه و تحلیل می شود و اطلاعات برای مستندسازی استخراج می گردند.
 - 3- بهبود ساختار برنامه: کنترل ساختار برنامه برای سهولت خواندن و فهمیدن تجزیه و تحلیل و اصلاح می شود.
 - 4- پیمانه بندی برنامه: قسمت های مرتبط برنامه با هم گروه بندی و قسمت های زائد حذف می شوند.
 - 5- مهندسی مجدد داده ها: داده ها توسط برنامه ی تغییر یافته برای انعکاس تغییرات برنامه پردازش می شوند.

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم



شکل 1: فرآیند مهندسی مجدد

3-3- مزایای کلیدی مهندسی مجدد سیستم های نرم افزاری

مهندسی مجدد سیستم نرم افزاری برای تکامل سیستم دو مزیت کلیدی نسبت به رویکردهای اصلی دیگر دارد:

- ریسک کم: در توسعه ی مجدد ریسک زیادی نهفته است که برای یک سازمان ضروری است. خطاها ممکن است در تشخیص سیستم، مسائل توسعه یافته و ... روی دهد.
- هزینه ی کاهش یافته: هزینه ی مهندسی مجدد نشان از کم بودن نسبت به هزینه ی توسعه ی سیستم جدید است.

4- طبقه بندی انواع مهندسی مجدد نرم افزار

در این بخش انواع مهندسی مجدد نرم افزار ارائه خواهد شد:

4-1- مهندسی رو به جلو یا مهندسی پیشرو (FE¹)

مهندسی پیش رو روش سنتی با حرکت به سمت انتزاع سطح بالا و اجرای طراحی های مستقل به منظور اتمام فیزیکی سیستم می باشد

4-2- مهندسی معکوس (RE²)

مهندسی معکوس جزء کلیدی نرم افزار مهندسی مجدد است که هر فعالیتی را برای درک نرم افزار آماده یا بهبود می دهد، معمولاً برای افزایش نگهداری، استفاده مجدد یا تکامل استفاده می شود.

4-3- مستندسازی مجدد³

مستندسازی مجدد یک تجدید نظر یا ایجاد یک تصویر وابسته به معنایی در سطح مشابه از انتزاع است. این نیز قدیمی ترین و ساده ترین فرم مهندسی معکوس است [12].

4-4- طراحی معکوس یا بازیابی طراحی⁴

طراحی معکوس یا بازیابی طراحی ترکیبی از اسناد موجود، کد، تجربه شخصی، دامنه های کاربردی و اطلاعات عمومی در مورد مساله دوباره ایجاد می کند.

4-5- درک برنامه⁵

1 Forward Engineering
2 Reverse Engineering
3 Re-documentation
4 Reverse design
5 Program comprehension

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

درک برنامه در ارتباط با فرآیند مهندسی معکوس است. درک برنامه به این معنی است که درک با برنامه شروع می شود، در حالی که RE ممکن است در یک ساختار اجرایی و باینری سیستم یا در توصیف های پیشرفته طراحی سیستم شروع شود [13].

۵-۴- بازسازی^۱

بازسازی فرآیند تبدیل داده ها از یک ساختار نمایی به سطح مشابه دیگر در سطح انتزاعی بسیار مشابه است و در عین حال حفاظت از معناشناسی و عملکرد سیستم نرم افزار انجام می شود [14].

۷-۴- کدسازی مجدد^۲

فرآیند کدسازی مجدد شامل تغییر ویژگی های برنامه است. کنترل جریان بازسازی و ترجمه زبان، تغییرات سطح برنامه هستند [15].

۸-۴- طراحی مجدد^۳

طراحی مجدد فرآیند تغییر ویژگی های طراحی است. منظور از تغییرات طراحی شامل بهبود الگوریتم، معماری طراحی بازسازی، تغییر داده ها از یک سیستم مدل در یک پایگاه داده و یا در ساختارهای داده ها می باشد [16].

۵- تفاوت مهندسی پیشرو و مهندسی مجدد

فرآیند مهندسی مجدد اغلب یک محرک برای تکامل نرم افزار است. چون سیستم های قدیمی ممکن است وابستگی های مجازی را با فرآیندهای موجود ترکیب کنند. پس قبل از اینکه فرآیند مهندسی مجدد اجرا شود، باید آن ها را کشف کرد و از بین برد. بنابراین وقتی مقیاس تغییرات توسط مهندسی مجدد فرآیند تجارت توانایی تطبیق با نگهداری برنامه ی نرمال را ندارد، نیاز به مهندسی مجدد نرم افزار در شرکت بیش از پیش آشکارتر می شود .

تمایز بین مهندسی مجدد و توسعه ی نرم افزار جدید نقطه ی شروعی برای پیشرفت است که این نقطه ی شروع با ویژگی های سیستم قدیمی به عنوان یکی از ویژگی های سیستم جدید می باشد .

چیکوفسکی و کراس^۴ توسعه ی متعارف و قراردادی را مهندسی پیشرو نامیدند تا آن را از مهندسی مجدد نرم افزار تمیز بدهند. مهندسی پیشرو با ویژگی های سیستم شروع می شود و طراحی و پیاده سازی سیستم جدید را درگیر می کند ولی مهندسی مجدد با سیستم موجود شروع می کند [17].

جدول ۱: ویژگی های مهندسی پیشرو

جدول ۲: ویژگی های مهندسی مجدد

عنوان	ویژگی ها
-------	----------

1 Restructuring

2 Recode

3 Redesign

4 Chikofsky and Cross

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

عنوان	ویژگی ها		
مهندسی پیشرو	مهندسی پیشرو	حرکت به سوی نیازمندیها برای طراحی یا حرکت به سوی طراحی برای پیاده سازی.	تعریف
		بخشی از مهندسی مجدد است	
		در مهندسی معکوس انجام می شود	
		یک نوع مهندسی نرم افزار است	
		اضافه کردن هیچ ویژگی های جدید برای کاربران	نباید ها
		بخشی از مهندسی نرم افزار است	نگهداری
		پیش بینی دقیق اینکه چقدر زمان برای توسعه	چالش
		سیستم یا مجموعه ای از تغییرات خاص نیاز دارد	
		انجمنی که شامل فعالیت های سیستمی درگیر در طراحی، پیاده سازی و آزمایش نرم افزار برای بهینه سازی و پشتیبانی آن	تعریف استاندارد کانادین
		برنامه سیستمی با راهکار منظم برای توسعه، عملیات و نگهداری نرم افزار است.	تعریف IEEE
برنامه مهندسی نرم افزار است.			
حل مسائل مشتریان است	هدف	مهندسی پیشرو	مهندسی پیشرو
تضمین اینکه نگهداری و تکامل نرم افزار توسط یک روش سیستماتیک انجام می شود.	بخشی از		
تکامل، نگهداری، مدیریت پروژه های نرم افزاری، برنامه نویسی، مدیریت پروژه.			
تکنیک های شناخته شده در سازماندهی و راه و روش دیسیپلین			
ترجمه طراحی ها سطح بالا به زبان های برنامه نویسی خاص	شامل		
نیرومند است.	ویژگی ها		
بسیار تکراری است.			
در حال توسعه در تکنولوژی و تکنیکها است.			
به طور معمول برای سازماندهی پروژه های نرم افزاری استفاده می شود			
گاهی اوقات شامل توسعه نرم افزار کاملا جدید می شود			
منابع مانند زمان و پول ذینفعان و زمان پردازنده و حافظه کامپیوترها	کاربران		
اصلاح نرم افزاری است که قبلا نوشته شده به طوری که نرم افزار معمولا به طور مداوم در طول یک سال تغییر می کند تا اینکه نسخه قبلی منسوخ شود.	معمولا شامل		
یک نوع نگهداری برای بهبود طراحی بخشی از یک سیستم نرم افزاری است که قابلیت نگهداری بالاتر را دارد. به طور کلی، هیچ ویژگی جدیدی برای کاربران اضافه نشده است	تعریف	مهندسی مجدد	مهندسی مجدد
برای افزایش قابلیت نگهداری یک سیستم است	هدف		
مهندسی پیشرو را دارد.			
دوباره ایجاد کردن یا اصلاح کردن (refactoring)	بخش ها		
• تمیز کردن کد را برای خواندن بیشتر • به طور کامل جایگزین یک لایه • اصلاح کردن (دوباره ایجاد کردن) بخش طراحی	شامل		
تعمیر و نگهداری است.	نوع		
مهندسی نرم افزار است			
هزینه های دراز مدت را کاهش می دهد	باعث کاهش		
اضافه کردن ویژگی های اجتناب ناپذیر سیستم آینده	بایدها		

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

اضافه کردن هیچ ویژگی های جدید برای کاربران	نباید ها	
بخشی از مهندسی نرم افزار است	نگهداری	

۶- تفاوت طراحی مجدد و مهندسی مجدد

دو مفهوم طراحی مجدد^۱ و مهندسی مجدد بعضی اوقات به جای یکدیگر بکار می روند اما این دو مفهوم تفاوت های زیادی با همدیگر دارند که شامل مورد زیر است:

در مهندسی مجدد تغییرات به صورت ناگهانی و ریشه ای ولی در طراحی مجدد به صورت تدریجی می باشد و در مهندسی مجدد کارها به صورت بنیادی انجام می شود اما در طراحی مجدد اصلاح انجام می شود. در مهندسی مجدد فرآیند جدید ایجاد می شود اما در طراحی مجدد فرآیند فعلی را آسان می کند. در مهندسی مجدد کاملاً به تکنولوژی وابسته است اما در طراحی مجدد ممکن است به تکنولوژی وابسته باشد. در مهندسی مجدد درصد ریسک آن بالاست اما در طراحی مجدد درصد ریسک آن کم یا متوسط است. مهندسی مجدد مزایای زیادی دارد اما در طراحی مجدد مزایای کمی دارد.

۷- تفاوت مهندسی معکوس و مهندسی پیشرو و مهندسی مجدد

در این بخش به مقایسه مهندسی معکوس و پیشرو و مجدد پرداخته شده است.

جدول ۴: مقایسه مهندسی معکوس و پیشرو و مجدد

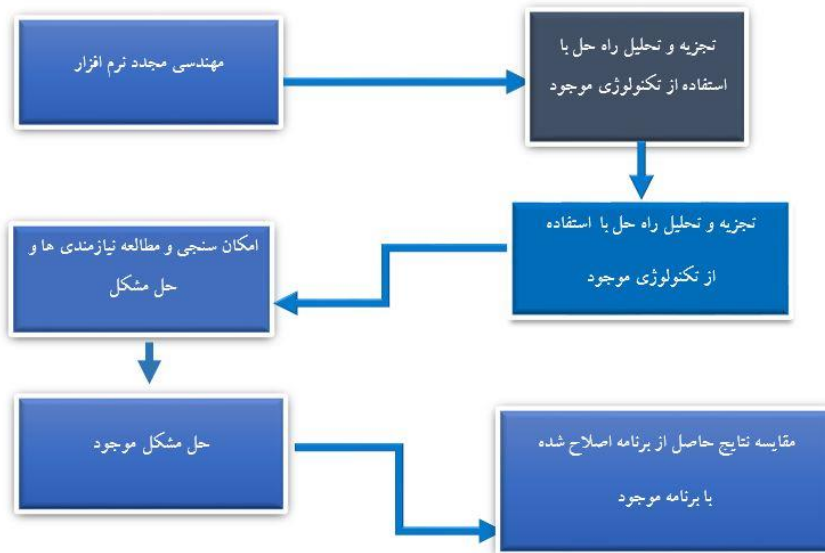
مهندسی معکوس	مهندسی پیشرو	مهندسی مجدد
این روش انتزاعی پاتین تر را به بالاترین تبدیل می کند.	روند سنتی حرکت از انتزاع سطح بالا و منطقی است. پیاده سازی وابسته به طراحی های پیاده سازی فیزیکی یک سیستم است	مهندسی مجدد محصول نهایی را بررسی و دوباره آن را می سازد اما به صورت بهتر
معمولاً برای بهبود سیستم استفاده می شود.	تغییرات در سیستم انجام می شود. مانند: ۱- استفاده از زبان برنامه نویسی متفاوت ۲- تولید یک DBMS جدید ۳- تبدیل چارچوب s/w به h/w جدید	معمولاً زمانی اجرا می شود که اسناد مناسب موجود نیست یا اسناد گم شده است.
بعد از پیدا کردن اینکه بیند محصول چطور کار می کند آن توسط یک تکنولوژی جدید یا متودولوژی جدید اجرا یا پیاده سازی می شود.	در مهندسی پیشرو یک طراحی برای یک ماشین انجام می شود تا کارهایی را انجام شود.	مهندسی معکوس راهی برای پیدا کردن اینکه چطور یک محصول نهایی کار می کند.
مهندسی مجدد به عبارتی دیگر یک قطعه جدید از نرم افزار را ایجاد می کند با وابستگی مشابه به آن چیزی که الان موجود است اما ممکن است شما آن را از طریق بهبود دهید.	مهندسی پیشرو یک مهندسی معمولی است. آن دستگاههایی را برای ما می تواند بسازد که واقعاً مفید باشند مانند: پل ها، کوره ها، ماشین ها و هواپیما ها.	مهندسی معکوس سعی می کند تا یک کد منبع را از کد کامپایل شده ایجاد کند. همچنین سعی می کند تا ترسیم کند که اجزای نرم افزار چطور در سیستم نهایی کار می کند.
مهندسی مجدد به سادگی یک بازسازی مجدد محصول است البته با مهندسی بهتر از قبل.	معمولاً شامل تغییر نرم افزاری است که نوشته شده است و این به دلیل این است که نرم افزار به مرور زمان تغییر می کند و منسوخ می شود.	مهندسی معکوس بخشی از محصول نهایی را با هدف آموزش نحوه کار کردن آن در بر می گیرد.

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

هر فعالیتی که نیازمند فهم برنامه در هر سطحی از محدوده ای از مهندسی معکوس که در آن هستیم را شامل می شود.	-	مهندسی مجدد ممکن است برای تغییرات کد یا نرم افزار مفید باشد.
---	---	--

۸- مکانیسم کارآمد و پیشرفته مهندسی مجدد نرم افزار

مکانیسم کارآمد و پیشرفته مهندسی مجدد نرم افزار یک فرآیند مهندسی مجدد مهندسی است که برای تبدیل سیستم نرم افزاری سنتی به یک سیستم نرم افزاری جدید مورد استفاده قرار می گیرد. شکل ۱، ساختار کلی مکانیسم بازسازی پیشنهاد شده ما را نشان می دهد. در ابتدا، سیستم ما اشکالاتی از شرایط موجود را از لحاظ دسترسی به زمان برنامه کاربردی و حجم ذخیره سازی نرم افزار مورد تجزیه و تحلیل قرار می دهد. پس از آن، راه حل را برای غلبه بر نقص سیستم موجود درون تکنولوژی پیاده سازی شده، تجزیه و تحلیل کنید. سپس امکان سنجی برای بهبود عملکرد سیستم موجود را بررسی کنید. سپس امکان سنجی برای حل مشکل موجود را بررسی کنید. در نهایت، نتیجه محصول اصلاح شده با محصول موجود را با توجه به زمان دسترسی و استفاده از حافظه مقایسه کنید. شکل ۲ سیستم پیشنهادی کلی را نشان می دهد [18].



شکل ۲: معماری سیستم مهندسی مجدد کارا و پیشرفته

۸-۱- روش کار سیستم مهندسی مجدد کارا و پیشرفته

چارچوب پیشنهادی شامل سه مرحله عمده می شود. مانند:

۱. مفهوم سیستم موجود را تحلیل کنید
۲. راه حل را با تکنولوژی موجود تجزیه و تحلیل کنید
۳. امکان سنجی و مطالعه نیازمندی ها و حل مشکل
۴. نتایج برنامه اصلاح شده را با برنامه موجود مقایسه کنید

۸-۱-۱- تحلیل مفهوم سیستم موجود

فرآیند اولیه از مکانیسم مهندسی مجدد م کارآمد و پیشرفته، تجزیه و تحلیل مفهوم سیستم موجود است. در این مرحله، توسعه دهنده مفهوم سیستم موجود و نیازهای سیستم مانند پس زمینه و پیش زمینه و مشخصات سخت افزاری را تحلیل می کند. به طور خلاصه، در ابتدا نیازهای فعلی یا نیازهای مشتری را دوباره مشخص کنید. علل شکست سیستم نرم افزاری

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

انتخاب شده در مرحله ۱ شناسایی شد، چندین سیستم نرم افزاری هیچ نوع امنیت و یا برخی از اشکالات دیگر را ندارند؛ فرآیند مدیریت خطا ضعیف وجود دارد که تمام موارد خطا احتمالی را پوشش نمی دهد. پس از تجزیه و تحلیل موارد فوق، توسعه دهنده باید عملکرد نرم افزار موجود مانند زمان پردازش و استفاده از حافظه و غیره را بداند.

۸-۱-۲- تجزیه و تحلیل راه حل با استفاده از تکنولوژی موجود

در این مرحله جزئیات در مورد تجزیه و تحلیل یافتن راه حل برای غلبه بر نقص در تکنولوژی پیاده سازی شده و روند تست دوباره توصیف می شود. قبل از پیدا کردن راه حل، شناسایی مشکل موجود یا وضعیت فعلی تکنولوژی موجود، روند بسیار مهمی در مهندسی مجدد نرم افزار است. وضعیت کنونی مهندسی مجدد سنتی شامل عملی بودن و قابلیت نگهداری است. مجموعه معیارهای نرم افزاری باید برای کمک به تشخیص مسائل کیفیت با سیستم نرم افزاری موجود و اولویت بندی برنامه هایی که متقاضیان مهندسی مجدد نرم افزار بر اساس ارزش کسب و کار و کیفیت فنی خود هستند، انتخاب شود. علاوه بر این، تبدیل تکنولوژی فعلی کد منبع به فناوری جدید نیز فرآیند پیچیده است؛ به طور متناوب، کارشناسان راه حل بهینه را با فن آوری فعلی که هزینه مصرف و مصرف حافظه را کاهش می دهد، پیدا می کند.

۸-۱-۳- امکان سنجی و مطالعه نیازمندی ها و حل مشکل

در این مرحله جزئیات در مورد امکان سنجی راه حل توضیح داده می شود. پس از پیدا کردن راه حل بهینه برای مهندسی مجدد نرم افزار موجود بدون تغییر تکنولوژی کنونی، امکان بررسی راه حل را بررسی می کند که مشکل بزرگی است. برای بررسی امکان سنجی، موارد زیر در نظر گرفته می شود: افزایش کیفیت نرم افزار، بهبود کارایی فرآیند نگهداری و افزایش ارزش کسب و کار. هنگام ایجاد چشم انداز، کاربر باید به روش قابل سنجش بیان شود مانند کاهش هزینه در بهبود عملکرد، حفظ مهندسی و کاهش عملیات و غیره. پس از آن، هزینه های بازسازی باید با صرفه جویی هزینه قابل پیش بینی و افزایش ارزش از نرم افزار کاربردی فعلی انجام گردد.

۸-۱-۴- مقایسه نتایج حاصل از برنامه اصلاح شده با برنامه موجود

این مرحله مرحله نهایی مکانیسم مجدد مهندسی کارآمد و پیشرفته است. بر اساس نتایج مراحل قبلی مهندسی نرم افزار، پیاده سازی یک برنامه نرم افزاری یا سیستم باید انجام شود. در پیاده سازی، یک بخش خاص ممکن است جایگزین بهترین آن شود که به طور کامل بر سه مرحله قبل از مکانیسم مجدد مهندسی کارآمد و پیشرفته ما متکی است. در نهایت، عملکرد نرم افزار اصلاح شده با نرم افزار کاربردی موجود با ویژگی هایی مانند زمان پردازش و استفاده از حافظه مقایسه می شود.

۱۱- نتیجه گیری

در این مقاله ابتدا به تعاریف مهندسی مجدد و معکوس و فرآیند این دو و در ادامه به مدل های آنها اشاره و در انتها به تفاوت آن دو اشاره شد مهندسی معکوس مشابه یک کپی برداری و نقشه برداری درحالی که مهندسی مجدد یک برداشت نوین درمدیریت در رابطه با تغییر فرآیند فعالیت های سازمان است. مهندسی معکوس مشابه تلاش به حل مسئله ای که از قبل رویش خلاقیت صورت گرفته ولی مهندسی مجدد با خلاقیت و ابتکار تغییر فرآیند سیستم را فراهم می کند. مهندسی معکوس نیازمند نمونه است وازنمونه به فرآیند می رسد اما مهندسی مجدد از فرآیند به نمونه می رسد. مهندسی مجدد نیازمند بهبود مستمر است در صورتی که مهندسی معکوس این چنین نیست. مهندسی معکوس نیاز به یک تیم چند تخصصی دارد اما مهندسی مجدد نیاز به همکاری همه ی افراد سازمان دارد. معمولا در مهندسی مجدد فرهنگ سازمانی تغییر می کند. اما در مهندسی معکوس معمولا این چنین نیست.

۱۲- مراجع

۱. شهیدی، محمدنقی، انتقال تکنولوژی و صنعتی کردن کشورهای در حال توسعه، دانشگاه تهران، ۱۳۷۲.

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

۲. عقلائی، اصغر. کریمی، رسول. جعفرنژاد، احمد، ارزیابی تطابق مدل های مهندسی معکوس با حوزه های معکوس با حوزه های دانشی مدیریت پروژه، فصلنامه علمی-ترویجی مدیریت استاندارد و کیفیت، شماره ۳، پاییز ۱۳۹۳، صفحات ۶۸-۸۷

۳. انصاری. علی، فرآیند مهندسی معکوس، پایان نامه ی کارشناسی رشته مهندسی مکانیک گرایش طراحی جامدات، بهمن ۱۳۷۹.

۴. محمودآبادی. علیرضا، تدوین فرآیند مهندسی معکوس در صنایع هوایی، پایان نامه کارشناسی ارشد رشته مهندسی صنایع گرایش مدیریت سیستم ها و بهره وری، دانشگاه صنعتی شریف، زمستان ۱۳۸۴.

۵. موحدی. مسعود، فرآیند طراحی معکوس، مجموعه مقالات اولین کنفرانس مهندسی ساخت و تولید در صنایع سنگین، دانشگاه صنعتی امیرکبیر، زمستان ۱۳۷۲، صفحات ۷۶۳-۷۷۳.

6. Concurrent Technologies Corporation, "Spare parts Reverse Engineering", 1998.
7. The Article of Daniel Gjorwell, Staffan Haglund, Daniel Sandell [BPR99] Jan.K.Collins, Business .Process Reengineering: AUSC Perspective, University South Carolina 1999.
8. Tamara J.Taylor, Frank Sparks, Building a Phased, Structured Approach to Reengineering Legacy Software Systems, September 1994.
9. Marchewka .T. Jack, "Information Technology Project Management", 2nd Edition, John Wiley & sons, 2006
10. A.INGLE, KATHRYN, "REVERSE ENGINEERING", MCGRAW- HILL, 1994.
11. Rick Dwar et al, Identifying and communicating expertise in systems reengineering: a patterns approach, June 1, 1999.
12. Tahvildari, L., Kontogiannis, K. On the role of design patterns in quality-driven re-engineering Proceedings of the IEEE 6th European Conference on Software Maintenance and Re-engineering (CSMR). 2002. Hungary.
13. M.Solvin, and S. Malik. "Re-engineering to reduce system maintenance: A case study", Software Engineering, pp.14-24, 2011.
14. Harry.M.Sneed, "Economicsof Software engineering", Journal of Software Maintenance, Vol.3, 1991, p.163.
15. S. Ducasse, T.G.1., and J.-M. Favre, Modeling software evolution by treating history as a first class entity, in on Software Evolution Through Transformation 2004. p. 71-82.
16. Moghaddas, Y.& Rashidi, H. (2009). A novel approach for replacing legacy systems, Journal of Applied Sciences, 9(22), 4086-4090.
17. E. J. Chikofsky and J. H. Cross II, "Reverse engineering and design recovery: A taxonomy", Index Technology Corp. and Northeastern University and Auburn university, (1990).
18. A. Cathreen, M.Chidambaram, EESRM: An Effective Approach to Improve the Performance of Software Re-Engineering, International Journal of Applied Engineering Research, (2018) pp. 3648-3654.