|  |
| --- |
| بسمه تعالی  نام ، نام خانوادگی: محمد مقری اسم استاد: استاد احمد فراهی  موضوع : مهندسی نرم افزار و مدل های مختلف آن شماره دانشجویی: 980198506 رشته: مهندسی کامپیوتر اسم درس: مهندسی نرم افزار |

مهندسی نرم‌افزار ( Software engineering) چیست؟

برای درک بهتر از مهندسی نرم افزار بهتر است از هم تفکیک شوند. این اصطلاح از دو کلمه نرم افزار و مهندسی ساخته شده است:

* نرم افزار چیزی بیش از کدهای اجرایی یک برنامه نیست. به طور کامل‌تر نرم افزار به مجموعه‌ای از کدهای برنامه نویسی اجرایی به همراه کتابخانه‌های مرتبط و اسناد مخصوص آن گفته می‌شود.
* از طرف دیگر مهندسی به همه موضوعات در مورد نحوه تولید محصولات، با استفاده از اصول و روش‌های علمی تعریف شده گفته می‌شود.
* موسسه استاندارد IEEE مهندسی نرم افزار را استفاده از یک روش منظم و قابل اندازه گیری برای توسعه، بهره برداری و نگهداری نرم افزار تعریف می‌کند.

مهندسی نرم افزار عبارت است از کاربرد مهندسی برای طراحی نرم افزار ،توسعه ، پیاده سازی و نگهداری از نرم افزار در یک روش سیستماتیک. یعنی استفاده از اصول مهندسی بجا و مناسب برای تولید و ارائه محصول نرم‌افزاری با کیفیت که قابل اطمینان و با صرفه بوده و بر روی ماشین‌های واقعی به‌طور کارآمدی عمل کند.

مزایای مهندسی نرم افزار :

بزرگترین مزیت مهندسی نرم‌افزار ایجاد یک مسیر ایمن برای طراحی و ارائه یک نرم افزار کامل، کاربردی و بدون مشکل است. مهندسی نرم افزار در کنار ارائه‌ی مدل‌ها و راه‌حل‌های مختلف روش‌هایی را جهت سهولت نگهداری، تعمیر و توسعه آن ارائه می‌دهد. علاوه بر این در کنار مزایای ذکر شده مزایای فوق العاده دیگری را به همراه می‌آورد که در ادامه مورد بررسی قرار خواهیم داد.

* پیچیدگی را کاهش میدهد

نرم افزارهای بزرگ همیشه پیچیده و دشوار هستند. مهندسی نرم افزار برای کاهش پیچیدگی هر پروژه یک راه حل عالی دارد. مهندسی نرم افزار مشکلات بزرگ را به چندین مشکل کوچک تقسیم می کند تا به راحتی شروع به حل هر یک از مشکلات کوچک کنید. تمام این مشکلات کوچک به طور مستقل از یکدیگر حل می شوند.

* به حداقل رساندن هزینه نرم افزار

برخی از نرم افزارها به سخت افزار زیادی احتیاج دارند و این هزینه‌ی کاربران نهایی را برای بهره بردن از آن‌‌ها افزایش می‌دهد. یا الگوریتم‌های نادرست استفاده شده در آن‌ها سبب می‌شود نیروی انسانی زیادی برای توسعه نرم افزار جهت تولید میلیون‌ها خط کد به کار گرفته شود. اما در مهندسی نرم افزار، برنامه نویسان همه چیز را برنامه ریزی می‌کنند و تمام موارد اضافی را که لازم نیست، کاهش می‌دهند و کدها را به صورت کاملا بهینه سازی شده به اتمام می‌رسانند.

همچنین اشتباهات حاصل از خطاها زمان و هزینه زیادی را در پی خواهد داشت که هزینه تولید را افزایش خواهد داد. اما به دلیل کاهش خطا و ارائه راه‌حل‌های علمی، هر نرم افزاری که از روش مهندسی نرم افزار استفاده می‌کند، هزینه کمتری را به افراد مرتبط با نرم افزار تحمیل می‌کند.

* کاهش زمان

هر چیزی که طبق برنامه ساخته نشده باشد، همیشه وقت را تلف می‌کند. اگر در حال تهیه نرم افزارهای بزرگ هستید بالطبع برای به دست آوردن کد اجرایی نهایی، نیاز به برنامه نویسی هزاران خط کد خواهید داشت. این فرآیند از شروع پروژه تا رسیدن به کد نهایی بسیار وقت گیر است و اگر به خوبی مدیریت نشود می‌تواند زمان زیادی را به خود صرف کند. بنابراین اگر شما نرم افزار خود را مطابق رویکرد مهندسی نرم افزار تهیه کنید، میزان زیادی از این هدر رفت زمان را کاهش خواهید داد.

* انجام راحت تر پروژه های بزرگ

پروژه های بزرگ چند روزه ساخته نمی‌شوند و به صبر، برنامه ریزی و مدیریت زیادی احتیاج دارند. مهندسی نرم افزار به جهت برنامه ریزی، جهت دهی به کارها، بخش بندی، تقسیم کار و آزمایش و نگهداری را در ارائه نرم‌افزارها سبب مجزا شدن بخش‌های مختلف و اجرای همروند آن‌ها می‌شود. بنابراین برای مدیریت پروژه‌های بزرگ شرکت‌های نرم افزاری باید به رویکرد مهندسی نرم افزار روی بیاورند.

* نرم افزار قابل اعتماد

نرم افزار باید قابل اعتماد باشد، به این معنی که اگر مهندسین نرم افزار آن را به شما تحویل داده‌اند، باید حداقل برای مدت زمان مشخص به درستی کار کند. و اگر اشکالی در نرم افزار وجود دارد، شرکت مهندسی مسئولیت حل همه این اشکالات را بر عهده دارد. از آنجا که در مهندسی نرم افزار، روش‌های آزمایش و نگهداری ارائه شده است، بنابراین هیچ نگرانی از قابلیت اطمینان بودن نرم افزارها وجود ندارد.

* اثربخشی

اثربخشی نشان می‌دهد که تا چه میزان از نتایج مورد نظر مشتری حاصل شده‌ است. اگر کارها مطابق با استانداردها انجام شده باشد، اثربخشی به وجود می‌آید. استانداردهای نرم افزاری تمرکز بزرگ شرکت‌ها برای اثربخشی بیشتر نرم افزار هستند. بنابراین نرم افزار با کمک این مباحث کاربردی عملکرد موثرتری خواهد داشت.

اهمیت مهندسی نرم افزار :

مهندسی نرم افزار بسیار مهم است زیرا تقریباً در هر صنعت، هر تجارت و برای هر عملکردی نرم افزار خاصی مورد نیاز است. و این موضوع با گذشت زمان از اهمیت بیشتری نیز برخوردار می‌شود. به عنوان مثال اگر در هنگام خرید آنلاین مشکلی در سبد خرید شما رخ دهد یک مشکل حیاتی است که بر وجهه آن فروشگاه آنلاین تاثیر می‌گذارد. بنابراین تمامی تلاش صاحبان کسب و کار استفاده از نرم افزارهایی با کمترین میزان خطا و مشکل است که این موضوع توسط تولید نرم افزار بر اساس اصول مهندسی نرم افزار امکان پذیر است.

در تولید صنعتی نرم افزار نیز تمایل به پیروی از یک فرآیند نرم افزاری جهت تضمین تولید یک محصول نهایی قابل تعمیر و نگهداری، قابل اعتماد و کاربردی است. از انتخاب زبان ب[**رنامه نویسی**](https://www.hamyarit.com/programing/computer-programming/) مناسب و الگوریتم‌ها گرفته تا در نظر گرفتن پویایی گروه، در دسترس بودن منابع و رضایت مشتریان همه مواردی هستند که مهندسان نرم افزار باید برای هر پروژه در نظر بگیرند و کیفیت حاصل از این اصول تمام چیزی است که کاربران نهایی آن نرم افزار نیاز خواهند داشت.

پیشینه مهندسی نرم افزار :

اصطلاح مهندسی نرم‌افزار پس از سال ۱۹۶۸ میلادی شناخته شد. این اصطلاح طی نشست «مهندسی نرم‌افزار ناتو ۱۹۶۸» (که در [گارمیش-پارتنکیرشن](https://fa.wikipedia.org/wiki/%DA%AF%D8%A7%D8%B1%D9%85%DB%8C%D8%B4-%D9%BE%D8%A7%D8%B1%D8%AA%D9%86%DA%A9%DB%8C%D8%B1%D8%B4%D9%86)، [آلمان](https://fa.wikipedia.org/wiki/%D8%A2%D9%84%D9%85%D8%A7%D9%86) برگزار شد) توسط ریاست نشست [فریدریش ال باوئر](https://fa.wikipedia.org/wiki/%D9%81%D8%B1%DB%8C%D8%AF%D8%B1%DB%8C%D8%B4_%D8%A7%D9%84_%D8%A8%D8%A7%D9%88%D8%A6%D8%B1) معرفی شد و از آن پس به‌طور گسترده مورد استفاده قرار گرفت.

اصطلاح مهندسی نرم‌افزار عموماً به معانی مختلفی به‌کار می‌رود:

* به‌عنوان یک اصطلاح غیررسمی امروزی برای محدوده وسیع فعالیت‌هایی که پیش از این [برنامه‌نویسی](https://fa.wikipedia.org/wiki/%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C) و [تحلیل سامانه‌ها](https://fa.wikipedia.org/wiki/%D8%AA%D8%AD%D9%84%DB%8C%D9%84_%D8%B3%D8%A7%D9%85%D8%A7%D9%86%D9%87%E2%80%8C%D9%87%D8%A7) نامیده می‌شد.
* به‌عنوان یک اصطلاح جامع برای تمامی جنبه‌های عملی برنامه‌نویسی رایانه، در مقابل تئوری برنامه‌نویسی رایانه، که [علوم رایانه](https://fa.wikipedia.org/wiki/%D8%B9%D9%84%D9%88%D9%85_%D8%B1%D8%A7%DB%8C%D8%A7%D9%86%D9%87) نامیده می‌شود.
* به‌عنوان اصطلاح مجسم‌کننده طرفداری از یک رویکرد خاص نسبت به برنامه‌نویسی رایانه که اصرار می‌کند، مهندسی نرم‌افزار، به‌جای آنکه هنر یا مهارت باشد، باید به‌عنوان یک رشته عملی مهندسی تلقی شود و از جمع‌کردن و تدوین روش‌های عملی توصیه‌شده به شکل متدولوژی‌های مهندسی نرم‌افزار طرفداری می‌کند.

محدوده مهندسی نرم افزار و تمرکز آن :

مهندسی نرم‌افزار به مفهوم توسعه و بازبینی یک سامانه نرم‌افزاری مربوط می‌باشد. این رشته علمی با شناسایی، تعریف، فهمیدن و بازبینی خصوصیات مورد نیاز نرم‌افزار حاصل سر و کار دارد. این خصوصیات نرم‌افزاری ممکن است شامل پاسخگویی به نیازها، اطمینان‌پذیری، قابلیت نگهداری، در دسترس بودن، آزمون‌پذیری، استفاده آسان، قابلیت حمل و سایر ویزگی ها باشد.

مهندسی نرم‌افزار ضمن اشاره به خصوصیات فوق، مشخصات معین طراحی و فنی را آماده می‌کند که اگر به‌درستی پیاده‌سازی شود، نرم‌افزاری را تولید خواهد کرد که می‌تواند بررسی شود که آیا این نیازمندی ها را تأمین می‌کند یا خیر.

مهندسی نرم‌افزار همچنین با خصوصیات پروسه توسعه نرم‌افزاری در ارتباط است. در این رابطه، با خصوصیاتی مانند هزینه توسعه نرم‌افزار، طول مدت توسعه نرم‌افزار و ریسک‌های توسعه نرم‌افزار درگیر است.

زیر شاخه های مهندسی نرم افزار :

* شناخت: بررسی و استخراج نیازمندی‌های نرم‌افزار که شامل استخراج، تحلیل و اعتبارسنجی خواسته‌ها و تهیه مستندات مربوط جهت طراحی نرم‌افزار می‌باشد.
* طراحی نرم‌افزار: فرایند تعریف معماری، اجزا، واسط و دیگر مشخصه‌های سیستم یا اجزا را گویند. همچنین این زیر بخش به عنوان خروجی فرایند نیز تعریف می‌شود.
* طراحی نرم‌افزار :توجهات طراحی، سازگاری، توسعه پذیری، تحمل خطا، قابلیت نگهداری، ماژولار بودن، قابلیت اطمینان، قابلیت استفاده مجدد، استحکام، امنیت، قابلیت استفاده، کارایی، قابلیت حمل، مقیاس پذیری.
* ساخت نرم‌افزار: جزئیات مربوط به ایجاد کار با معنی برای نرم‌افزار از طریق برنامه‌نویسی، بازنویسی و تأیید، تست واحدها و اجزا، تست یکپارچگی، و اشکال یابی.
* تست نرم‌افزار: بررسی فنی \_تجربی، برای ارائه به سهامداران و ذی نفعان که اطلاعاتی در مورد کیفیت محصول یا خدمات تحت تست بیان می‌کند.
* تعمیر و نگهداری نرم‌افزار: کلیه فعالیت‌های مورد نیاز برای ارائه پشتیبانی مقرون به صرفه در نرم‌افزار است.
* مدیریت پیکربندی نرم‌افزار:شناسایی پیکربندی یک سیستم در نقاط مشخصی از زمان به منظور کنترل سیستماتیک تغییرات پیکربندی و حفظ و نگهداری یکپارچگی برنامه و ردیابی پیکربندی در طول چرخه عمر سیستم را گویند.
* مدیریت نرم‌افزارهای مهندسی: نرم‌افزار مدیریت فعالیت‌ها و برنامه‌ریزی، هماهنگی، اندازه‌گیری، نظارت، کنترل و گزارش‌گیری به منظور حصول اطمینان از توسعه و نگهداری سیستماتیک، منضبط و اندازه‌گیری شونده نرم‌افزار است.
* فرایند نرم‌افزار: تعریف، اجرا و پیاده‌سازی، ارزیابی، اندازه‌گیری، مدیریت، ایجاد تغییر و بهبود فرایند چرخه حیات خود نرم‌افزار را گویند.
* روش‌های مهندسی نرم‌افزار و مدل‌های تحمیل ساختار در مهندسی نرم‌افزار با هدف سیستماتیک و منظم کردن فعالیت، قابل تکرار بودن و در نهایت افزایش کیفیت نرم‌افزار و موفقیت آمیزتر بودن آن ایجاد می‌شود.
* تمرین مهندسی نرم‌افزار حرفه‌ای درباره دانش، مهارت و نگرش‌های مهندسی نرم‌افزار است که مهندسان نرم‌افزار باید تمرین مهندسی نرم‌افزار را به صورت حرفه‌ای، مسئولانه و اخلاقی بکنند.
* اقتصاد مهندسی نرم‌افزار در مورد تصمیم‌گیری در زمینه کسب و کار تجاری مهندسی نرم‌افزار است.
* مبانی ریاضی و محاسباتی
* مبانی مهندسی
* ابزار مهندسی نرم‌افزار و روش‌ها: ابزارهایی مبتنی بر کامپیوتر برای مهندسی نرم‌افزار ایجاد شده‌اند تا به فرایندهای چرخه حیات نرم‌افزار و روش‌هایی که ساختاری را بر فعالیت‌های مهندسی نرم‌افزار اعمال می‌کنند کمک کند تا به هدف ساخت فعالیت‌های سیستماتیک و در نهایت به موفقیت بتوان رسید.

تست نرم افزار :

[تست نرم‌افزار](https://fa.wikipedia.org/wiki/%D8%AA%D8%B3%D8%AA_%D9%86%D8%B1%D9%85%E2%80%8C%D8%A7%D9%81%D8%B2%D8%A7%D8%B1) یکی از بخش‌های مهم و تأثیرگذار در مهندسی نرم‌افزار است. بالابردن کیفیت نرم‌افزارهای تولیدی بدون توجه به فرایند تست و تضمین کیفیت نرم‌افزار ممکن نخواهد بود. امروزه پیشگامان صنعت نرم‌افزار به این نتیجه رسیدند که توسعه و تست نرم‌افزار باید به صورت موازی با هم پیش روند. این امر سبب می‌شود که همراه با توسعه نرم‌افزار کیفی آن نیز تضمین شود. توجه به تست نرم‌افزار نکته متمایزکننده شرکت‌های موفق آی تی و شرکت‌های شکست خورده است.

**مهندسی نرم‌افزار یک رشته استراتژیک** و پولساز در سراسر دنیا است. البته رویکرد این رشته «اقتصاد تولیدمحور» است و وقتی تولید انبوه نرم‌افزارهای باکیفیت و «مهندسی‌ساز» هدف نباشد، همان «مهارت برنامه‌نویسی» نیز جواب‌گوی بسیاری از نیازها است. بعضی کشورهای در حال توسعه مانند هند و چین توجه ویژه‌ای به این موضوع دارند و سرمایه‌گذاری‌های فراوانی روی تقویت پایه‌های **مهندسی نرم‌افزار** در اقلیم خود می‌کنند. در ایران نیز این سرمایه‌گذاری روی توانمندسازی متخصصان این رشته می‌تواند به همان اندازه مفید باشد زیرا ابزارهای تولید در این رشته نسبت به سایر رشته‌های مهندسی ارزان‌تر و قابل بازیافتند.

منبع : hamyarit.com

wikipedia.org