



نرم‌افزار یک مفهوم انتزاعی و غیرقابل لمس است که با واژه مهندسی نرم‌افزار ارتباط مستقیمی دارد. مهندسی نرم‌افزار یک نظم مهندسی است که تمرکزش بر طراحی مقرون‌به‌صرفه سیستم‌های نرم‌افزاری با کیفیت بالا است.

این مطلب یکی از مقالات پرونده ویژه «متدولوژی‌ها، الگوها و معماری نرم‌افزار» شماره 207 ماهنامه شبکه است. علاقه‌مندان می‌توانند کل این پرونده ویژه را از روی سایت شبکه [دانلود](#) کنند.

اصطلاح مهندسی نرم‌افزار نخستین بار در سال 1968 در کنفرانس «بحران نرم‌افزار» مطرح شد. این بحران پس از آن‌که مدارهای مجتمع در دهه 60 میلادی اختراع شدند، به وجود آمد. توان محاسباتی قدرتمند مدارهای مجتمع راه را برای ساخت نرم‌افزارهای قدرتمندی که به مراتب پیچیده‌تر از قبل بودند، هموار کرد. این پیچیدگی به سرعت نشان داد که ساخت نرم‌افزارهای پیچیده بدون اتکا بر روش‌های ضابطه‌مند امکان‌پذیر نبوده، عملکرد پایین، ناپایداری زیاد، زمان‌بر بودن و هزینه‌هایی به مراتب فراتر از پیش‌بینی‌های اولیه را به همراه دارد. پیچیدگی بیش‌ازحد نرم‌افزارها نیز فرآیند حفظ و نگهداری آن‌ها را به مراتب دشوارتر از قبل کرده بود. ترکیب این عوامل با یکدیگر به شکل‌گیری بحران نرم‌افزار منجر شدند. درحالی‌که هزینه ساخت‌افزارها به سرعت رو به کاهش بود در مقابل هزینه تولید نرم‌افزارها روند صعودی به خود گرفته بود. این افزایش هزینه‌ها نشان داد، شرکت‌ها به تکنیک‌ها و روش‌های جدیدی برای کنترل و مهار پیچیدگی ذاتی موجود در سامانه‌های نرم‌افزاری بزرگ نیاز دارند. رویکردی که در نهایت به ابداع تکنیک‌های خلاقانه‌ای منجر شد که ما امروزه از اصلاح مدل‌ها، متدولوژی‌ها و معماری نرم‌افزار برای توصیف آن‌ها استفاده می‌کنیم. در شرایطی که متدولوژی‌های نرم‌افزاری در سراسر جهان به شدت موردتوجه شرکت‌ها و حتی عامه مردم قرار دارد (افرادی که با فناوری در ارتباط هستند)، در مقابل در داخل دیدگاه متفاوتی حاکم است. بسیاری از مردم بر این باور هستند که این تنها دانشجویان رشته‌های مهندسی نرم‌افزار و فناوری اطلاعات هستند که باید با متدولوژی‌های نرم‌افزار آشنایی داشته باشند و ضرورتی ندارد افرادی که به‌نوعی با دنیای فناوری در ارتباط هستند با چنین مفهومی آشنا باشند، اما این دیدگاه درست نیست. تاکنون چندیار سفارش نرم‌افزاری داده‌اید و پس از گذشت مدت‌زمانی با کاستی‌هایی در آن روبه‌رو شده‌اید و از شرکت سازنده درخواست کرده‌اید قابلیت‌هایی به نرم‌افزار ساخته‌شده اضافه کند اما آن‌ها این کار را با تاخیر انجام داده یا درخواست هزینه بیشتری کرده‌اند؟



قدیمی اما همچنان اثرگذار چرا مدل آبشاری هنوز جزو برترین متدولوژی‌های دنیای نرم‌افزار است؟

پیش از آن‌که سفارش ساخت نرم‌افزاری را به شرکتی بدهید، بدون توجه به بزرگی یا کوچکی یک شرکت، ابتدا باید از آن‌ها سوال کنید که قرار است نرم‌افزار مدنظر شما را بر مبنای چه متدولوژی یا مدل توسعه نرم‌افزاری طراحی کنند. این یک پرسش کلیدی است، به دلیل آن‌که اگر نرم‌افزار شما منطبق با اصول مهندسی نرم‌افزار و متدولوژی‌های نرم‌افزاری نوشته‌شده باشد، شما همراه با نرم‌افزاری که دریافت خواهید کرد، مستندات مربوط به ساخت آن‌را نیز دریافت می‌کنید. حتی اگر نرم‌افزار مدنظر شما یک سایت است، این مستندات هستند که در آینده به شما اجازه خواهند داد بر مبنای نقشه راهی که ترسیم‌شده قابلیت‌هایی را به سایت خود اضافه کرده، تغییراتی در آن اعمال کرده یا مواردی را حذف کنید. نکته مهم دیگری که لازم است به آن توجه داشته باشید نوع متدولوژی است که یک شرکت برای ساخت نرم‌افزار شما از آن استفاده می‌کند. زمانی که از شرکتی در این ارتباط سوال کردید، هر واژه‌ای که در ارتباط با ساخت نرم‌افزار از زبان مدیر آن شرکت بیان شود، به سبک ساخت نرم‌افزار شما اشاره خواهد داشت. اسکرام، فرآیند توسعه شخصی، برنامه‌نویسی مفرط، توسعه سریع، نمونه‌سازی اولیه، پیچشی و... هر یک سبک‌های خاصی از تولید نرم‌افزار هستند که برخی بر ساخت سریع یک نرم‌افزار، برخی بر مدیریت دقیق ریسک‌ها و برخی بر کیفیت بالای نرم‌افزارها تاکید دارند. شرکتی که نرم‌افزار مدنظر شما را بدون اتکا بر هیچ متدولوژی نرم‌افزاری طراحی می‌کند، درست همانند فردی کار می‌کند که در نظر دارد بدون هیچ‌گونه دانش فنی یا اصول معماری یک ساختمان بسازد. طبیعی است در چنین شرایطی اگر نرم‌افزار موردنیاز شما با مشکل روبه‌رو شود، پیدا کردن مشکل یا تغییر در سورس کدها کاری بسیار دشوار خواهد بود. سفارش ساخت سایت که این روزها به‌شدت محبوب شده مثال خوبی در این زمینه است. شرکت از طریق قرار دادن چند ماژول ساده در کنار یکدیگر و کمی کدنویسی جاوااسکریپت و CSS سایتی به شما تحویل می‌دهد که مملو از کاستی‌ها است. سایتی که رابط کاربری آن مشکل داشته، رنگ‌ها به درستی انتخاب نشده‌اند، سایت در ارتباط با مباحث سئو داخلی با مشکل روبه‌رو بوده و در عمل حداقل بهینه‌سازی ممکن روی کدها انجام شده است. واضح است که چنین شرکتی از کم اطلاعی مشتری سوءاستفاده کرده، بدون توجه به اصول مهندسی نرم‌افزار و به‌کارگیری متدولوژی‌ها، نرم‌افزاری ارائه کرده و فقط با مشتری قراردادی منعقد کرده است که در زمان نگهداری و خدمات پس از فروش مبلغی از مشتری دریافت کرده و یکسری سرویس‌های اولیه به او ارائه کند. اما این تنها مردم عادی نیستند که به‌واسطه عدم آشنایی با متدولوژی‌ها متضرر می‌شوند. افرادی که در حوزه کاری خود صاحب تخصص هستند و به‌خوبی می‌دانند چطور باید از یک‌زبان برنامه‌نویسی یا یک ابزار طراحی استفاده کنند، به‌واسطه عدم آشنایی با متدولوژی‌های نرم‌افزاری به راحتی نمی‌توانند شغل مناسب خود را پیدا کنند. امروزه، در بسیاری از آگهی‌های استخدامی شما در کنار تخصص‌های فنی واژگان دیگری همچون اسکرام، چابک و... را مشاهده می‌کنید.

مطلب پیشنهادی



طراحی و توسعه نرم‌افزارها به شکل V بایدها و نبایدهای به‌کارگیری مدل V در ارتباط با توسعه نرم‌افزارها

این واژگان بیانگر آن هستند که شیوه کار شرکت و تیم‌های برنامه‌نویسی آن‌ها چگونه است. به‌عنوان مثال اگر با متدولوژی‌های نرم‌افزاری آشنا نباشید و به استخدام شرکتی درآمده باشید و در جلسه تیم توسعه‌دهندگان مدیر پروژه درخواست نمونه‌سازی افقی یا عمودی کنید، شما تنها می‌توانید به چشمان او نگاه کنید، به دلیل این‌که نمی‌دانید این اصطلاحات چیستند و به چه مفهومی اشاره دارند. همین چند خط به‌خوبی بیانگر این موضوع است که مبحث

متدولوژی‌های نرم‌افزار و معماری نرم‌افزار تا چه اندازه مهم بوده و موضوعاتی نیستند که تنها برنامه‌نویسان باید به آن‌ها اشراف دقیق داشته باشند. بر همین اساس در این پرونده تصمیم گرفتیم شما را با تعدادی از متدولوژی‌ها و مدل‌های معروفی که در دنیای نرم‌افزار وجود دارد آشنا کنیم. نکته‌ای که لازم است در انتهای این مقدمه به آن اشاره داشته باشیم این است که ما در این پرونده به یکسری متدولوژی‌های پرکاربرد همچون فرآیند نرم‌افزار شخصی، توسعه افزایشی، سریع و... اشاره نکردیم؛ به دلیل این‌که در شماره‌های گذشته مجله شبکه در بخش کارگاه این متدولوژی‌ها را موردبررسی قرار داده‌ایم.

برای دانلود پرونده ویژه «متدولوژی نرم‌افزار» اینجا کلیک کنید
تاریخ انتشار:

28 دی 1397

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/programming/13816/%D8%A7%DA%AF%D8%B1-%D9%85%D9%87%D8%A7%D8%B1%D8%AA-%D8%AF%D8%A7%D8%B1%DB%8C%D8%AF-%D8%A7%D9%85%D8%A7-%D8%B4%D8%BA%D9%84%DB%8C-%D9%86%D8%AF%D8%A7%D8%B1%DB%8C%D8%AF%D8%8C-%D9%85%D8%AA%D8%AF%D9%88%D9%84%D9%88%DA%98%DB%8C-%D9%86%D8%B1%D9%85%E2%80%8C%D8%A7%D9%81%D8%B2%D8%A7%D8%B1-%D8%B1%D8%A7-%D8%AF%D8%B1%DB%8C%D8%A7%D8%A8%DB%8C%D8%AF>