به نام خدا

فرایند مهندسی نرم افزار

چارچوب فرایند با تعیین تعداد کوچکی از فعالیت های چارچوبی که برای کلیه پروژه های نرم افزاری قابل استفاده باشند، صرف نظر از اندازه و پیچیدگی آنها، شالوده ای برای یک فرایند مهندسی نرم افزار کامل پی ریزی می کند. یک چارچوب فرایند کلی برای مهندسی نرم افزار شامل پنج فعالیت می شود.

ارتباطات : پیش از اینکه هرگونه کار فنی آغاز شود، برقراری ارتباط و همکاری با مشتری بسیار مهم است. هدف، درک اهداف طرف های ذی نفع برای پروژه و جمع آوری خواسته هایی است که می توانند ویژگی ها و قابلیت های عملیاتی نرم افزار را تعیین کنند.

برنامه ریزی : یک پروژه نرم افزاری، سفری پیچیده است و فعالیت برنامه ریزی، نقشه ای ایجاد می کند که به راهنمایی تیم در انجام این سفر کمک می کند. این نقشه با توصیف وظایف فنی که قرار است اجرا شوند، خطرات احتمالی، منابعی که مورد نیاز خواهند بود، محصولات کاری ای که باید تولید شوند و زمانبندی کاری، مهندسی نرم افزار را مشخص می کند.

مدل سازی : یک معمار، هر روز با مدل ها کار می کند، اِتودی می زند تا تصویر بزرگ را درک کند، اینکه از نظر معماری چه ظاهری دارد، بخش های سازنده اش چگونه با هم جور در خواهند آمد، و بسیاری خصوصیات دیگر. مهندسی نرم افزار با ایجاد مدل هایی جهت درک بهتر خواسته ها و طراحی که به این خواسته ها برسد، همین کار را می کند.

ساخت : این فعالیت، تولید کدها و آزمون لازم برای آشکار کردن خطاهای موجود در کدها را با هم تلفیق می کند.

استقرار : نرم افزار به مشتری تحویل داده می شود تا محصول تحویل داده شده را ارزیابی کرده و بر اساس این ارزیابی، بازخوردی ارائه دهد.

برای بسیاری از پروژه های نرم افزاری، فعالیت های چارچوبی به موازات پیشرفت پروژه به صورت تکراری به کار برده می شوند. در هر دور از تکرار پروژه، یک نسخه از نرم افزار ایجاد می شود که زیر مجموعه ای از قابلیت های عملیاتی و ویژگی های نرم افزار کامل را در اختیار افراد ذی نفع قرار می دهد. با تولید هر نمو، نرم افزار کامل و کامل تر می شود.

فعالیت های چارچوبی فرایند مهندسی نرم افزار توسط تعدادی از فعالیت های چتری تکمیل می شوند که عبارتند از:

کنترل و پیگیری پروژه های نرم افزاری: به تیم نرم افزاری امکان می دهد تا پیشرفت را در مقایسه با نقشه پروژه بسنجد و هر گونه کنش لازم را برای حفظ زمان بندی به عمل آورد.

مدیریت ریسک: خطراتی را ارزیابی می کند که ممکن است بر نتیجه پروژه یا کیفیت محصول تاثیر بگذارند.

تضمین کیفیت نرم افزار: فعالیت های لازم برای حصول اطمینان از کیفیت نرم افزار را معین می کند.

بازبینی فنی: محصولات کاری مهندسی نرم افزار را در تلاش برای آشکار کردن خطاها قبل از انتشار آنها در فعالیت بعدی و برطرف کردن آنها ارزیابی می کند.

اندازه گیری: موازینی از فرایند، پروژه و محصول را تعریف می کند که نیازهای طرف های ذی نفع را برطرف می سازند.

مدیریت پیکربندی نرم افزار: اثرات تغییرات را در سراسر فرایند نرم افزار مدیریت می کند.

مدیریت قابلیت استفاده مجدد: ملاک های مربوط به استفاده مجدد را تعریف می کند و سازوکارهایی برای دستیابی به قطعات قابل استفاده مجدد برقرار می سازد.

تهیه و تولید محصول کاری: شامل فعالیت های لازم برای ایجاد محصولات کاری از قبیل مدل ها، مستندات، وقایع نگارها ، فرم ها و فهرست ها می شود.

توجه به این نکته ضروری است که فرایند مهندسی نرم افزار یک دستورالعمل نهایی و غیر قابل تغییر نیست که تیم نرم افزاری باید با تعصب از آن پیروی کند بلکه باید سریع الانتقال و انطباق پذیر باشد.

بنابراین فرایندی که برای یک پروژه پذیرفته می شود، ممکن است با فرایند پذیرفته شده برای پروژه های دیگر تفاوتی چشمگیر داشته باشد. در مقالات بعدی مدل های مختلف فرایندها را شرح خواهیم داد.

مهندسی نرم افزار در عمل

جورج پولیا در یک کتاب کلاسیک جوهر حل مساله و در نتیجه در مهندسی نرم افزار را چنین مطرح می کند:

شناخت مساله

طرح ریزی برای یک حل

اجرای برنامه ریزی

بررسی نتیجه برای صحت

اصول کلی :

دیوید هوکر هفت اصل را مطرح نموده است که توجه به آنها در مهندسی نرم افزار بسیار ضروری به نظر می رسد:

اصل یکم) دلیل وجود سیستم: هر سیستم به یک وجود نیاز دارد: این که برای کاربرانش ارزش فراهم سازد. همه تصمیم گیری ها باید با مد نظر داشتن این نکته انجام شود.

اصل دوم) ساده نگه داشتن: همه طراحی ها باید تا حد امکان ساده باشند. این باعث می شود که یک سیستم قابل فهم تر با قابلیت نگهداری بالاتر را داشته باشید.

اصل سوم) حفظ چشم انداز: برای موفقیت یک پروژه نرم افزاری، چشم اندازی روشن، ضروری است و بدون آن پروژه تقریبا همواره به جایی می رسد که دو یا چند ایده بر آن حاکم خواهد شد. یک سیستم بدون یکپارچگی مفهومی، به مجموعه ی ناجوری از طراحی های ناسازگار تبدیل می شود که به یکدیگر وصله-پینه شده اند. مسامحه در مورد خصوص چشم انداز معماری یک سیستم نرم افزاری باعث تضعیف سیستمی با طراحی خوب و سرانجام از کار افتادن آن می شود.

اصل چهارم) آنچه که شما تولید می کنید، دیگران مصرف می کنند: همواره تعیین مشخصات، طراحی و پیاده سازی را طوری انجام دهید که دیگران نیز قادر به درک کار شما باشند.

اصل پنجم) آینده نگری: سیستمی با طول عمر بالا از ارزش بیشتری برخوردار است. سیستم ها باید آمادگی انطباق بر تغییرات را داشته باشند. سیستم هایی که این ویژگی ها را با موفقیت ارائه می دهند، از ابتدا با این ویژگی ها طراحی می شوند.

اصل ششم) برنامه ریزی پیشاپیش برای استفاده مجدد: استفاده مجدد باعث صرفه جویی در زمان و کار می شود. استفاده مجدد از کد ها و طراحی ها به عنوان مزیت اصلی فن آوری های شیئ گرا مطرح شده است ولی این امکان در برنامه نویسی شیئ گرا نیازمند برنامه ریزی قبلی است.

اصل هفتم) تفکر: این آخرین اصل احتمالاً بیش از بقیه مورد بی مهری قرار می گیرد. تعقل و تفکر کامل و روشن قبل از اقدام به عمل، همواره نتایج بهتری به بار می آورد. با تفکر روشن درباره سیستم، ارزش آن بالا می رود. به کارگیری شش اصل نخست نیاز به تفکر عمیق دارد و در این صورت، فایده بسیاری از آن عاید خواهد شد.

هانیه اکبری 980097175