

مهندسی نرم افزار ۱

فصل اول

دانشجو: آيسا تاجيك

استاد: آقاي احمد فرحي

► نرم افزار چیست؟

نرم افزار کامپیوتری، محصولی است که مهندس نرم افزار طراحی می کند و می سازد. شامل برنامه هایی می شود که در هر کامپیوتری به هر اندازه و با هر معماری، قابل اجرا هستند، مستنداتی دارد که شامل فرم های واقعی و مجازی می شود و داده هایی دارد که ترکیبی از ارقام و حروف است و البته می تواند شامل اشکال نمایشی از قبیل اطلاعات تصویری و ویدیویی و صوتی باشد.

► اما چه می کند؟

مهندسان نرم افزار آن را می سازند و در حقیقت هر کسی در دنیای صنعت چه مستقیم و چه غیر مستقیم از آن استفاده می کند

► چرا اهمیت دارد؟

چون تقریباً همهی جنبه های زندگی ما را تحت تأثیر قرار می دهد و در تجارت، فرهنگ و فعالیتهای ور روزمره ما نمایان است.

► چه مراحلی دارد؟

نرم افزارهای کامپیوتری نیز همانند تمام محصولات موفق دیگر ساخته می شوند، یعنی با اجرای فرایندی چابک و انعطاف پذیر، منجر به نتیجه ای با کیفیت بالا میشود و نیازهای کاربران آن را برآورده می سازد. شما روش مهندسی نرم افزار را به کار خواهید بست.

► در این محصول کار چیست؟

از دیدگاه مهندس نرم افزار، محصول کار، برنامه ها ، مستندات و داده ها هستند که نرم افزار کامپیوتری است. ولی از دیدگاه کاربر، محصول کار، اطلاعاتی است که به نحوی به درد کاربر می خورند.

► نرم افزار کامپیوتری همچنان مهمترین فن آوری در صحنه جهانی به شمار می رود. و در عین حال مثالی از قانون پیامدهای ناخواسته است. پنجاه سال قبل هیچ کس نمی توانست پیش بینی کند که نرم افزارها به یک فناوری ضروری برای تجارت، علوم و مهندسی تبدیل خواهند شد؛ که با کمک نرم افزار فناوری های جدیدی (مانند مهندسی ژنتیک و فن آوری نانو) خلق شود، فن آوری های موجود (مانند ارتباطات) بسط و توسعه یابند و در فن آوری های قدیمی تر (مانند صنعت چاپ تغییرات بنیادی پدید آید؛ که نرم افزارها نیروی محرکه ای برای انقلاب در زمینه کامپیوترهای شخصی شوند؛

که

مشتریان بتوانند از مغازه سرکوپه نرم افزارها را در قالب بسته بندی های کوچک خریداری کنند؛ که نرم افزار به آهستگی از یک محصول به سرویسی تکامل پیدا کند که شرکت های نرم افزاری مطابق با درخواست مشتری در قالب یک عملکرد ویژه از طریق مرورگر وب در اختیار او قرار دهند؛ که یک شرکت نرم افزاری از تقریباً هر شرکت صنعتی هم عصر خودش بزرگتر و تاثیر گذارتر شود، که یک شبکه گسترده که با نرم افزار هدایت می شود و به آن اینترنت گفته می شود، تکامل یابد و همه چیز را از جستجو در کتابخانه گرفته تا خرید از فروشگاه و مسائل سیاسی تغییر دهد. . .

هیچ کس قادر به این پیش بینی نبود که نرم افزارها در هر نوع سیستمی تعبیه خواهند شد: حمل و نقل، پزشکی، ارتباطات، نظامی، صنعتی، سرگرمی، ماشین های اداری... و این فهرست تقریباً پایانی ندارد. و اگر به قانون پیامدهای ناخواسته اعتقاد دارید، اثرات فراوانی وجود دارد که هنوز قابل پیش بینی نیست.

هیچ کس پیش بینی نمی کرد که میلیونها برنامه کامپیوتری را باید اصلاح کرد، تطبیق داد و با گذر زمان بهبود بخشید. زحمت اجرای این فعالیت های نگهداری از کل کار لازم برای ایجاد نرم افزار . جدید بیشتر است و به افراد و منابع بیشتری نیاز دارد

► ماهیت نرم افزاری

امروزه نرم افزار نقشی دوگانه دارد. نرم افزار نوعی محصول است و در عین حال وسیله ی نقلیه ای برای تحویل یک محصول به عنوان محصول، توان محاسباتی بالقوه ی یک سخت افزار کامپیوتری یا به طور . گسترده تر، شبکه ای از کامپیوترها را بالفعل می کند.

نرم افزار چه در داخل یک تلفن همراه باشد و چه درون یک کامپیوتر بزرگه عمل کند، یک مبدل اطلاعات است. تولید مدیریت، اکتساب، اصلاح نمایش با انتقال اطلاعاتی که می تواند به سادگی یک بیت باشد یا به پیچیدگی یک نمایش چندرسانه ای.

نرم افزار به عنوان وسیله ی نقلیه ای برای تحویل یک محصول، مبنای کنترل کامپیوتر (سیستم عامل)، مخابرات اطلاعات (شبکه ها) و خلق و کنترل برنامه های دیگر (محیطها و ابزارهای نرم افزاری) را تشکیل میدهد.

► نرم افزار جایی است که در آن رویا کاشته می شود و کابوس درو می شود، یک برکه اسرار آمیز و انتزاعی که در آن ارواح پلید برای اکسیرهای جادویی در رقابت اند، جهانی یک تعریف رسمی تر، احتمالا درک شما را به میزان محسوس تری افزایش خواهد داد. برای دستیابی به این منظور، بررسی خصوصیات نرم افزار که آن را با سایر ساخته های دست بشر متفاوت می سازد، حائز اهمیت است.

► نرم افزار، بیشتر یک عنصر منطقی است تا یک عنصر سیستمی فیزیکی بنابراین، نرم افزار دارای خصوصیتی است که تفاوت چشمگیری با سخت افزار دارد.

اگرچه شباهتهایی میان بسط
نرم افزار و ساخت سخت افزار
وجود دارد، این دو عمل، تفاوت
بنیادی دارند.

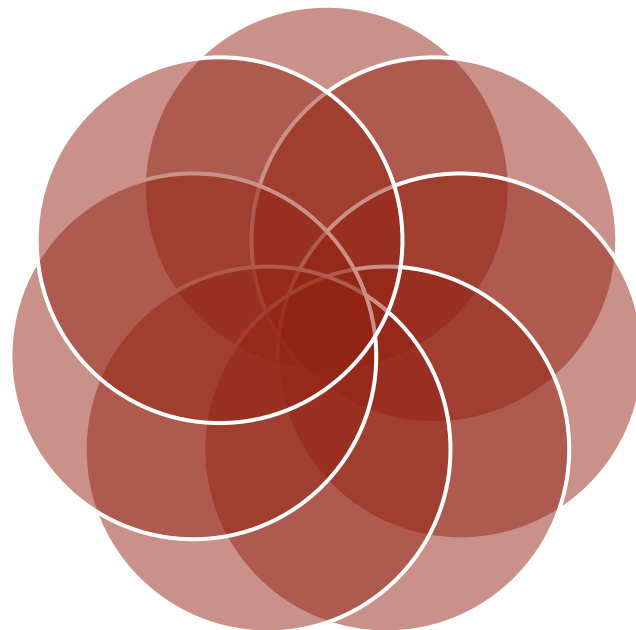
در هر دو عمل، کیفیت بالا از
طریق طراحی خوب به دست
می آید، ولی فاز ساخت برای
سخت افزار باعث بروز
مشکلات کیفیتی می شود
که برای نرم افزار وجود ندارند
یا به راحتی قابل رفع هستند.

هر دو عملی وابسته به
انسان هستند، ولی رابطه
میان انسان و کاری که انجام
می شود، کاملاً متفاوت
است

**نکته ی کلیدی : نرم
افزارها ساخته نمی
شوند، بلکه مهندسی
می شوند**

هزینه های نرم افزار در
مهندسی آن متمرکز است. این
بدان معناست که پروژه های
نرم افزاری را نمی توان همانند
پروژه های تولید معمولی
مدیریت کرد.

هر دو عمل مستلزم ساخت
یک محصول هستند ولی روش
ها متفاوت است.



. نرم افزار مهمترین محصول عصر ما را تحویل می دهد: اطلاعات.. نرم افزار داده های شخصی مانند تراکنش های مالی یک فرد را تبدیل می کند به طوری که به چیزهای مفیدتری در یک حیطه ی محلی تبدیل شوند؛ اطلاعات تجاری را مدیریت می کند تا رقابت را بهبود بخشد؛ دروازه ای است به سوی شبکه های اطلاع رسانی جهانی مانند اینترنت و وسیله ای است برای به دست آوردن اطلاعات در تمامی اشکال آن

نقش نرم افزارهای کامپیوتری طی ۵۰ سال گذشته دستخوش تغییرات فراوان شده است. پیشرفت های عالی در زمینه کارایی سخت افزار، تغییرات بنیادی در معماری کامپیوتر، افزایش زیاد .. حافظه و ظرفیت ذخیره سازی و انواع دستگاه های ورودی و خروجی، همگی در پیچیده تر شدن سیستم های کامپیوتری سهیم بوده اند. پیچیدگی سیستم می تواند باعث حصول نتایج درخشان شود ولی برای کسانی که قرار است سیستمی پیچیده را بسازند، مشکلات عظیمی به بار می آورد

• تعریف نرم افزار

امروزه اکثر حرفه ای ها و بسیاری از افراد عامه، سخت معتقدند که می دانند توصیفی درسی از نرم افزار می تواند به شکل زیر باشند نرم افزار عبارت است از:

► دستورالعمل ها برنامه های کامپیوتری که هنگام اجرا، ویژگی، عملکرد و کارایی مطلوب را فراهم می سازند

► ساختمانهای داده هایی که برنامه ها را قادر به پردازش مناسب داده ها کنند و اطلاعات توصیفی در هر دو قالب کپی سخت و مجازی که راه اندازی و استفاده از برنامه ها را : شرح دهند.....

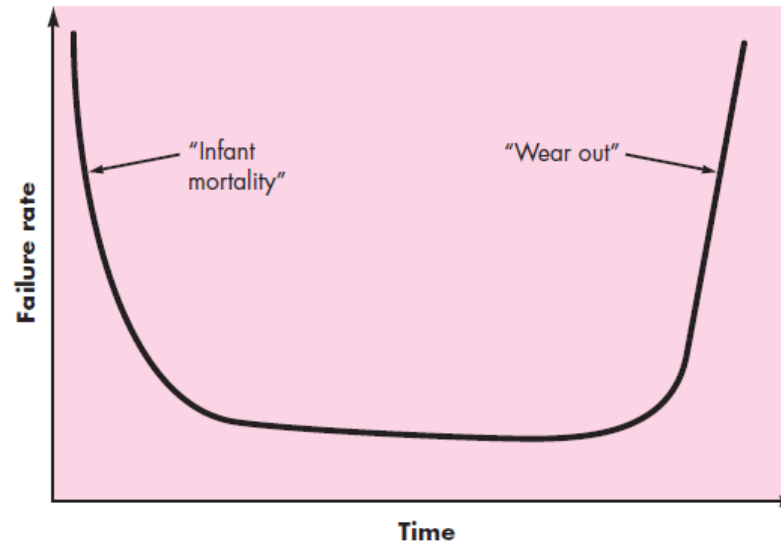
تردیدی نیست که تعاریف دیگری از نرم افزار نیز قابل ارائه است.

شکل ۱-۱ نمودار آهنگ شکست را به صورت تابعی از زمان برای سخت افزار نشان میدهد. این رابطه که غالباً **منحنی وانیه** نامیده می شود، نشان میدهد که سخت افزار، آهنگ شکست نسبتاً شدیدی در ابتدای عمر خود نشان می دهد این شکست ها را غالباً می توان به عیوب طراحی و تولید نسبت داد

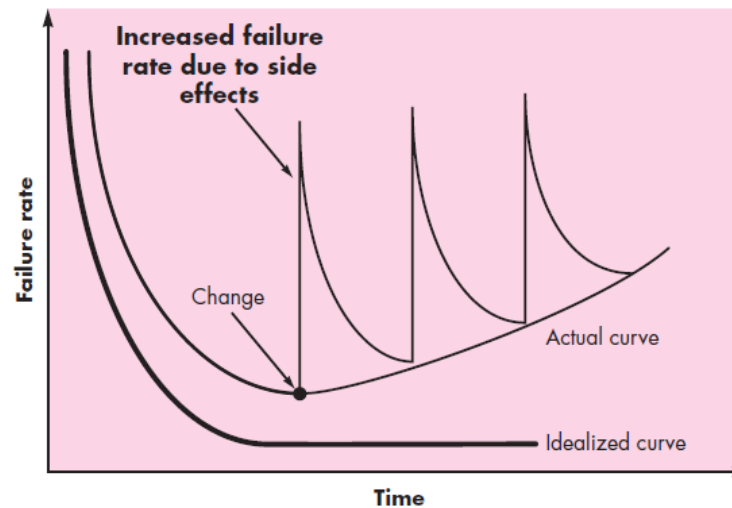
این عیوب تصحیح می شوند و آهنگ شکست برای یک دوره زمانی به مقداری ثابت نزول می کند (که امید می رود، بسیار پایین باشد)

. با گذشت زمان، سخت افزار شروع به فرسایش کرده دوباره آهنگ شکست شدت می گیرد.

نرم افزار نسبت به ناملایمات محیطی که باعث فرسایش آن می شود، نفوذپذیر نیست.



شکل ۱-۱



- ▶ این تناقض ظاهری را می توان با در نظر گرفتن منحنی واقعی به بهترین وجه توضیح داد (شکل ۱-۲). نرم افزار در دوران حیات خود دستخوش تغییر می شود (نگهداری).
- ▶ با اعمال این تغییرات، احتمال دارد که برخی عیوب جدید وارد شوند و باعث خیز منحنی آهنگ شکست شوند
- ▶ پیش از آنکه منحنی بتواند به آهنگ شکست منظم اولیه خود برسد، تغییر دیگری درخواست می شود که باعث خیز دوباره ی منحنی می شود.
- ▶ حداقل میزان شکست به آهستگی افزایش می یابد
- ▶ - نرم افزار در اثر تغییر فاسد می شود.
- ▶ یک جنبه ی دیگر از فرسایش نیز اختلاف میان سخت افزار و نرم افزار را نشان می دهد.
- ▶ هنگامی که یک مؤلفه از سخت افزار فرسوده می شود، با یک مؤلفه ی یدکی تعریف می شود. ولی نرم افزار قطعات یدکی ندارد.
- ▶ هر شکست نرم افزاری نشانگر خطایی در طراحی با فرایندی است که طراحی از طریق آن به گدهای قابل اجرا روی ماشین تبدیل می شود.
- ▶ از این رو، نگهداری نرم افزار به مراتب پیچیده تر از نگهداری سخت افزار است.

ماهیت منحصر به فرد برنامه های کاربردی تحت وب

در روزهای اولیه شبکه جهانی وب (حدود ۱۹۹۰ تا ۱۹۹۵)، وب سایتها شامل یک مجموعه فایل های آپرمتن پیونددار می شد که اطلاعات را با استفاده از متن و گرافیک محدود ارائه می دادند.

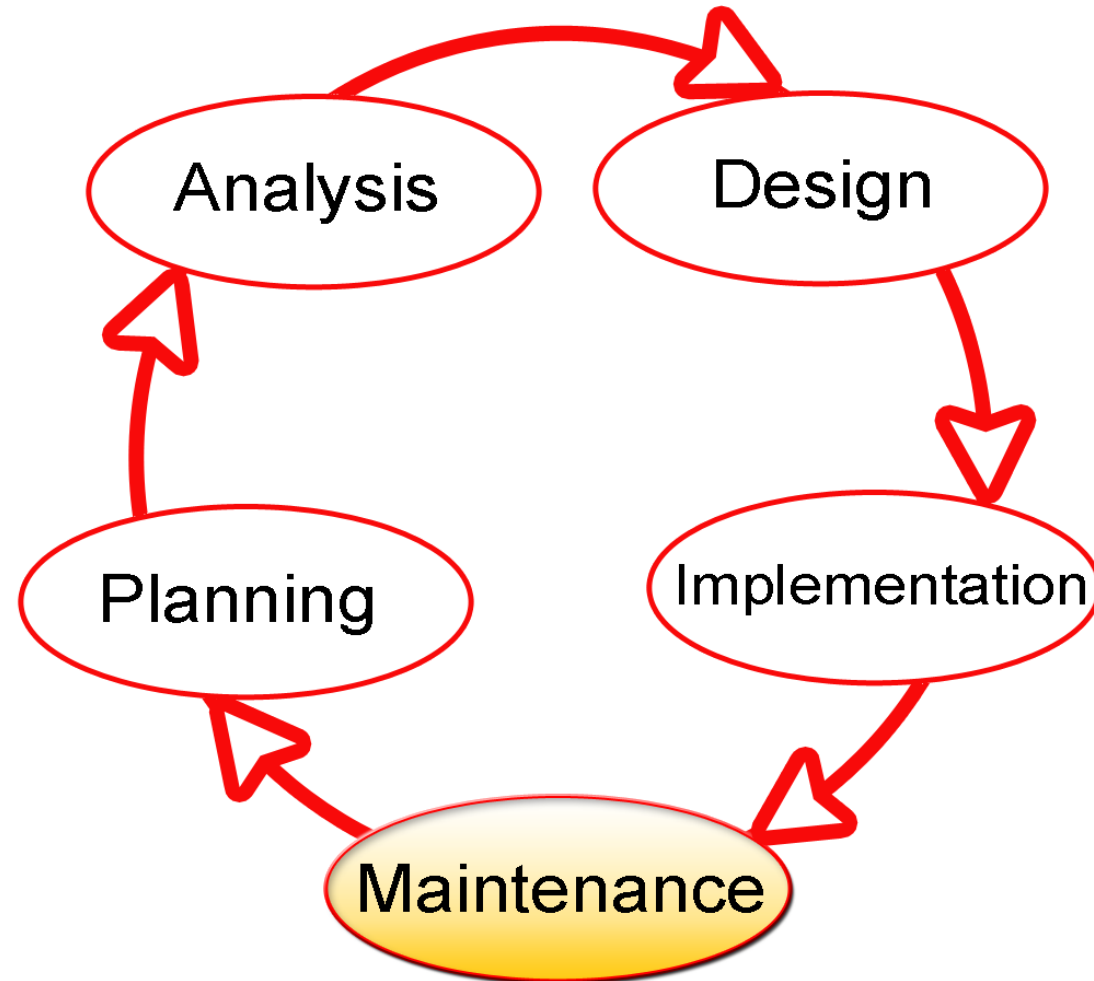
با گذشت زمان، تکمیل HTML با ابزارهای برنامه نویسی (مانند XML و جاوا) به مهندسان وب این امکان را داد تا قابلیت های کامپیوتری را همراه با محتوای اطلاعاتی فراهم سازند. سیستم ها و برنامه های کاربردی مبتنی بر وب (که آنها را در کل برنامه های تحت وب می نامیم) با به عرصه وجود نهادند.

امروزه برنامه های تحت وب به ابزارهای کامپیوتری پیچیده ای تکامل یافته اند که نه تنها عملکردی مستقل را و در اختیار کاربر نهایی قرار می دهند، بلکه با بانکهای اطلاعاتی و برنامه های کاربردی تجاری یکی شده اند.

برنامه های تحت وب، یکی از گروههای متمایز نرم افزارند. با این حال، می توان استدلال کرد که برنامه های تحت وب، متفاوتند.

۱. برنامه ی تحت وب شامل همه چیز از یک صفحه وب ساده می شود که ممکن است به مصرف کننده کمک کند تا بدهی خود را برای خرید خودرو پرداخت کند تا یک وبسایت جامع و فراگیر که خدمات مسافری کامل را برای بازرگانان و افراد عادی که به تعطیلات می روند، فراهم می آورد.
۲. این گروه شامل وبسایتهای کامل، عملکردهای خاص موجود در وبسایت ها و برنامه های پردازش اطلاعات میشود که روی اینترنت با روی یک اینترنت با اکسترانت قرار دارند.

Systems development life cycle



اصول کلی واژه ی اصل (principle) به معنای یک قانون یا فرض زیربنایی است که در یک سیستم فکری وجود آن ضروری است.

برخی از این اصول به مهندسی نرم افزار به عنوان یک کلیت توجه دارند و برخی دیگر به یک فعالیت چارچوبی کلی مشخص (مانند ارتباطات)، می پردازند و از آن گذشته عده ای نیز بر کنش های مهندسی نرم افزار مانند طراحی معماری با وظایف فنی مانند نوشتن یک سناریوی کاربرد تأکید دارند.

این اصول، در هر سطحی از تأکید که قرار داشته باشند، به شما در برقراری یک فضای فکری برای - مهندسی نرم افزار در عمل، کمک خواهند کرد و اهمیت آنها از این لحاظ است.

.پاول هفت اصل را مطرح نموده است که کانون توجه آنها کار در مهندسی نرم افزار به عنوان یک کلیت است. .. این اصول را در بندهای زیر عیناً تکرار می کنیم

اصل نخست

دلیل وجود سیستم و هر سیستم به یک وجود نیاز دارد. این که برای کاربرانش ارزشی فراهم سازد. همهی تصمیم

اصل دوم: ساده نگه داشتن

طراحی نرم افزار یک فرایند تصادفی نیست و در هر تلاش برای طراحی باید عوامل فراوانی را در نظر گرفت.

همه ی طراحی ها باید تا حد امکان ساده باشد ولی نه ساده تر.

این باعث می شود که داشتن یک سیستم قابل فهم تر با قابلیت نگهداری بالاتر آسان تر شود.

این بدان معنا نیست که ویژگی های سیستمی، حتی آنهایی که درونی هستند باید به بهانه ساده سازی کنار گذاشته شوند.

در واقع طراحی های ظریفتر معمولا طراحی های ساده ترند. ساده. در عین حال به معنی سریع و نامنظمه هم نیست. در حقیقت، ساده سازی نیاز به مقادیر متناهی فکر و کار در چند دور تکرار دارد.

نتیجه نرم افزاری با قابلیت نگهداری بالاتر و کم خطا خواهد بود.

اصل سوم

حفظ چشم انداز (vision) برای موفقیت یک پروژهی نرم افزاری، چشم اندازی روشن، ضروری است و بدون آن پروژه تقریباً همواره به جایی می رسد که دو یا چند ایده بر آن حاکم خواهد شد. یک سیستم بدون یکپارچگی مفهومی، به مجموعه ی ناچوری از طراحی های ناسازگار تبدیل می شود که به یکدیگر وصله پینه شده

اصل چهارم

آنچه که شما تولید می کنید، دیگران مصرف می کنند

اصل پنجم: آینده نگری

سیستمی با طول عمر بالا. از ارزش بیشتری برخوردار است. در محیطهای کامپیوتری امروزی، که مشخصات به طور لحظه ای تغییر می کنند و تنها با گذشت چند ماه، سکوها ی سخت افزاری، کهنه و . قدیمی به شمار می روند، طول عمر نرم افزارها به جای سال بر حسب ماه سنجیده می شود. ولی سیستم های صنعتی پرفدردت باید پیش از اینها دوام بیاورند. برای انجام موفقیت آمیز این کار، سیستم ها باید آمادگی انطباق بر این تغییرات و سایر تغییرات را داشته باشند. سیستم هایی که این ویژگی را با موفقیت ارائه می دهند، از ابتدا با این ویژگی طراحی می شوند. هرگز طوری طراحی کنید که خود را گرفتار کنید. همواره از خود بپرسید، «اگر فلان مورد پیش آید، چه خواهد شد؟»

اصل ششم

برنامه ریزی پیشاپیش برای استفاده مجدد استفاده ی مجدد باعث صرفه جویی در زمان و کار می شود. می توان استدلال کرد که دست بافتن به سطح بالایی از قابلیت استفاده ی مجدد، سخت ترین هدف در رسیدن به یک سیستم نرم افزاری است. استفاده ی مجدد از کدها و طراحی ها به عنوان مزیت اصلی فن آوری های شیءگرا مطرح شده 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. هنگامی که درباره چیزی فکر کنید، احتمال اینکه آن را درست انجام دهید، بیشتر می شود. به علاوه، درباره انجام دوبارهی آن اطلاعاتی به دست خواهید آورد. اگر درباره چیزی فکر کنید و هنوز آن را اشتباه انجام دهید، این به یک تجربه با ارزش تبدیل می شود. یک اثر جانبی تفکر، این است که پی می برید هنگامی که چیزی را نمی دانید، در کدام نقطه می توانید در جستجوی پاسخ باشید. 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.

**امیدوارم
مفید
بوده
باشه**