بسمه تعالی

درس مهندسی نرم افزار ۱

نیمسال اول ۹۳-۹۲

مرضیه سادات طباطبایی

فصل اول

نرم افزار و مهندسی نرم افزار

نگاهی گذرا

- نرم افزار چیست؟
  - نرم افزار، محصولی است که مهندس نرم افزار طراحی می کند و می سازد.
  - شامل برنامه، مستندات و داده ها می شود.
- چه می کند؟
  - مهندس نرم افزار آن را می سازد و هر کس دردنیای صنعت از آن استفاده می کند.
- چرا اهمیت دارد؟
  - چون همه جنبه های زندگی ما را تحت تاثیر قرار می دهد.

نگاهی گذرا

- چه مراحلی دارد؟
  - با اجرای فرایندی چابک و انعطاف پذیر مبتنی بر اصول مهندسی نرم افزار ساخته می شود.
- محصول کار چیست؟
  - از دیدگاه مهندس نرم افزار: برنامه ها، مستندات و داده ها
  - از دیدگاه کاربر: اطلاعاتی است که به نحوی به درد کاربر میخورد.

ماهیت نرم افزار

- امروزه نرم افزار نقشی دو گانه دارد:
  - نرم افزار نوعی محصول است.
  - در عین حال وسیله نقلیه ای برای تحویل یک محصول است.
- نرم افزار مهمترین محصول عصر ما را تحویل میدهد: اطلاعات
- نرم افزار عبارت است از:
  - دستورالعملها که هنگام اجرا، ویژگی، عملکرد و کارایی مطلوب را فراهم می سازند.
  - ساختمان های داده ایی که برنامه ها را قادر به پردازش مناسب داده ها کنند.
  - اطلاعات توصیفی در هر دو قالب کپی سخت و مجازی که راه اندازی و استفاده از برنامه ها را شرح دهند.

تفاوتهای نرم افزار و سخت افزار

- نرم افزار با سخت افزار تفاوت چشمگیری دارد:
  - نرم افزار، مهندسی و بسط داده می شود و چیزی نیست که به معنای کلاسیک کلمه ساخته شود.
  - نرم افزار فرسوده نمی شود.
  - گرچه صنعت در حال حرکت به سوی مونتاژ قطعات است، اکثر نرم افزارها همچنان به صورت سفارشی ساخته می شوند.

نرم افزار، مهندسی و بسط داده می شود

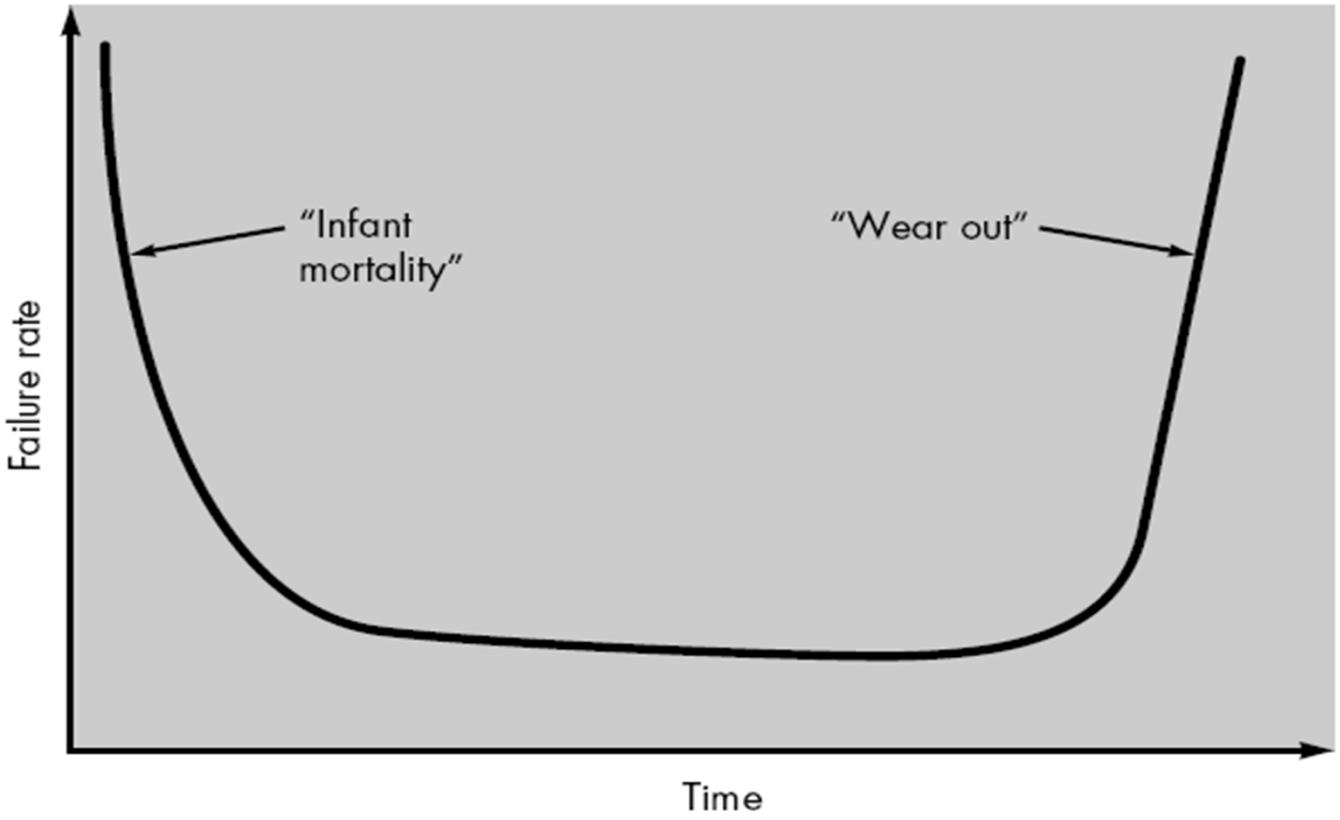
- تفاوت های بسط سخت افزار و نرم افزار:
  - فاز ساخت برای سخت افزار باعث بروز مشکلات کیفیتی می شود که برای نرم افزار وجود ندارند یا به راحتی قابل رفع است.
  - هر دو عمل وابسته به انسان هستند ولی رابطه ی میان انسان و کاری که انجام می شود، کاملا متفاوت است.
  - هر دو عمل مستلزم ساخت یک محصول هستند ولی روش ها متفاوت است.

نرم افزار فرسوده نمی شود.

- منحنی شکست سخت افزار
  - منحنی وانی نمودار آهنگ شکست را به صورت تابعی از زمان برای سخت افزار نشان می دهد.
  - منحنی وانی، نشان می دهد که سخت افزار در ابتدای عمر آهنگ شکست شدیدی دارد(عیوب طراحی و تولید).
  - این عیوب، تصحیح و آهنگ شکست در یک دوره زمانی به مقداری ثابت نزول می کند.
  - با گذشت زمان سخت افزار شروع به فرسایش کرده و دوباره آهنگ شکست شدت می گیرد.

نرم افزار فرسوده نمی شود.

- منحنی شکست سخت افزار



نرم افزار فرسوده نمی شود.

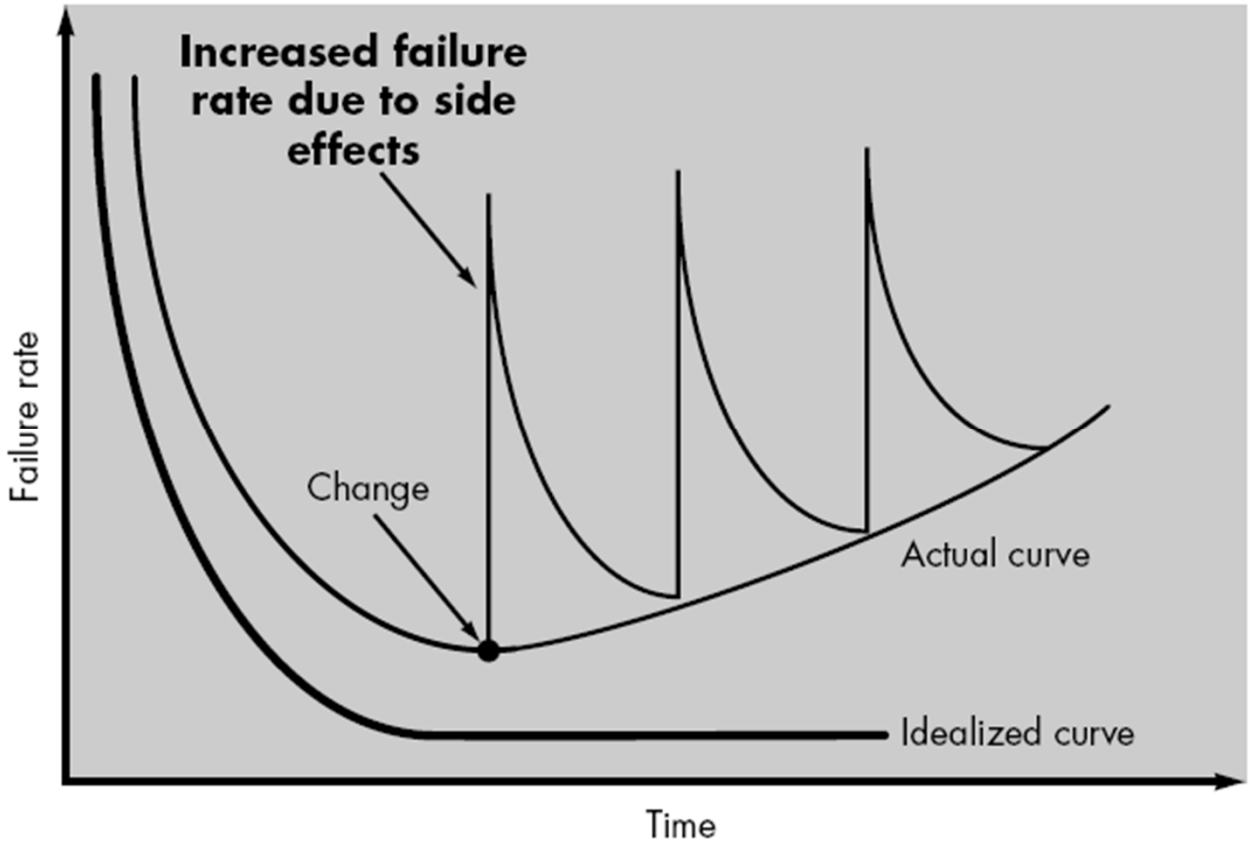
- منحنی های شکست واقعی و ایده آل نرم افزار
  - نرم افزار نسبت به ناملایمات محیطی که باعث فرسایش آن می شود نفوذ پذیر نیست.
  - بنابراین در تئوری، منحنی شکست برای نرم افزار باید شکل منحنی ایده آل را به خود بگیرد. عیوب کشف نشده باعث آهنگ شکست شدید در ابتدای عمر برنامه شده و با برطرف شدن این عیوب منحنی هموار می شود.

نرم افزار فرسوده نمی شود.

- منحنی های شکست واقعی و ایده آل نرم افزار
  - نرم افزار در دوران حیات خود دست خوش تغییر می شود.
  - با اعمال این تغییرات احتمال دارد که برخی عیوب جدید وارد شوند و باعث خیز منحنی آهنگ شکست می شود.
  - و پیش از آن که منحنی بتواند به آهنگ شکست منظم اولیه خود برسد، تغییر دیگری درخواست می شود که باعث خیز دوباره منحنی می شود.

نرم افزار فرسوده نمی شود.

- منحنی های شکست واقعی و ایده آل نرم افزار



نرم افزارها همچنان به صورت سفارشی ساخته می شوند.

- در جهان سخت افزار، استفاده ی مجدد از قطعات بخشی طبیعی از فرایند مهندسی است.
- در مهندسی نرم افزار این امر به تازگی مورد توجه قرار گرفته است.

دامنه های کاربرد نرم افزار

- نرم افزار های سیستمی
  - مجموعه ای از برنامه هاست که برای سرویس دهی به برنامه های دیگر نوشته شده اند.
- نرم افزارهای کاربردی
  - برنامه های مستقلی که یک نیاز تجاری مشخص را بر طرف می کند.
- نرم افزارهای مهندسی/علمی
  - نرم افزارهای علمی توسط الگوریتم هایی مشخص می شوند که اعداد و ارقام را پردازش می کنند. کاربردهای نوین در حیطه مهندسی و علمی از الگوریتم های عددی مرسوم فراتر رفته اند.

دامنه های کاربرد نرم افزار

- نرم افزارهای تعبیه شده
  - در حافظه فقط خواندنی جای دارند و برای کنترل محصولات و سیستم های مربوط به بازارهای صنعتی و مصرفی به کار می رود.
- نرم افزارهای خط تولید
  - برای فراهم آوردن یک قابلیت خاص جهت استفاده توسط بسیاری از مشتریان مختلف طراحی می شوند.
- برنامه های کاربردی تحت وب
  - این گروه از نرم افزارهای شبکه ای شامل مجموعه ی گسترده ای از برنامه های کاربردی می باشد.

دامنه های کاربرد نرم افزار

- نرم افزارهای هوش مصنوعی
  - برای حل مسائل پیچیده ای که به روش های عددی قابل حل نیستند، از الگوریتم های غیر عددی استفاده می کنند.
  - سیستم های خبره، تشخیص الگوها، شبکه های عصبی مصنوعی، اثبات قضایا و بازی مثال هایی از کاربرد این گروه هستند.

چالشهای پیش رو

- کار با کامپیوتر در جهانی باز
  - چالشی که مهندسان فرا روی خود خواهند داشت، توسعه ی سیستم ها و برنامه های کاربردی است که برقراری ارتباط میان کامپیوترهای شخصی، دستگاه های همراه و سیستم های اداری را از طریقه شبکه های گسترده میسر می سازند.
- تامین منابع از طریق شبکه
  - رشد سریع شبکه جهانی وب و لزوم معماری برنامه های کاربردی ساده و پیچیده برای استفاده کاربران نهایی در سراسر جهان
- کد منبع باز
  - تمایل رو به رشدی است که منجر به توزیع کدهای منبع سیستم ها و برنامه های کاربردی شده است به طوری که افراد بسیاری بتوانند در توسعه آن سهیم شوند.

نرم افزارهای قدیمی

- سیستم های نرم افزاری قدیمی چند دهه قبل ساخته شده اند و پیوسته اصلاح شده اند تا تغییرات به عمل آمده درخواست های تجاری و سکوهای محاسباتی را پاسخگو باشند.
- ازدیاد این گونه سیستم ها باعث دردسر برای سازمانهای بزرگی می شود که نگهداری از آن ها را پر هزینه و تکامل بخشیدن به آن ها را خطرناک می دانند.

دلایل تکامل سیستمهای قدیمی

- نرم افزار باید برای برآورده ساختن نیازهای محیطهای جدید کامپیوتری یا فناوری های جدید اصلاح گردد.
- نرم افزار باید بهبود یابد تا خواسته های تجاری جدید را پیاده سازی کند.
- نرم افزار باید گسترش داده شود تا با سایر سیستمها یا بانک اطلاعاتی جدیدتر، قابلیت همکاری را داشته باشد.
- نرم افزار باید دوباره معماری شود تا در یک محیط شبکه نیز قادر به ادامه ی حیات باشد.

ماهیت برنامه های کاربردی تحت وب

- امروزه برنامه های تحت وب به ابزارهای کامپیوتری پیچیده ای تکامل یافته اند که عملکردی مستقل را در اختیار کاربر نهایی قرار می دهند و با بانک های اطلاعاتی و برنامه های کاربردی تجاری یکی شده اند.
- در اکثریت وسیع برنامه های تحت وب مشخصه های اسلاید بعد مشاهده می شود.

ماهیت برنامه های کاربردی تحت وب

- میزان تمرکز شبکه
  - برنامه های تحت وب روی یک شبکه قرار دارند و باید نیازهای جامعه ای متنوع از کلاینت ها را برآورده سازند.
- همروندی
  - ممکن است یک باره تعداد بسیاری از کاربران به برنامه های تحت وب دستیابی داشته باشند.
- بار غیر قابل پیشبینی
  - تعداد کاربران برنامه های تحت وب از روزی به روز دیگر ده یا صد برابر میشود.

ماهیت برنامه های کاربردی تحت وب

- کارایی
  - اگر کاربر یک برنامه تحت وب باید یک مدت طولانی منتظر بماند، ممکن است تصمیم بگیرد به جای دیگری برود.
- قابلیت دسترسی
  - گرچه انتظار ۱۰۰ در صد قابلیت دسترسی، غیر منطقی است، کاربران برنامه های تحت وب پر طرفدار غالبا تقاضای دسترسی ۲۴ ساعته در ۷ روز هفته و ۱۲ ماه سال را دارند.

ماهیت برنامه های کاربردی تحت وب

- داده محوری
  - عملکرد اصلی بسیاری از برنامه های تحت وب استفاده از ابررسانه ها برای ارائه متون به کاربران نهایی است.
- حساس به محتویات
  - کیفیت و ماهیت زیبا شناختی محتویات از جمله مهمترین عوامل تعیین کننده کیفیت در برنامه های تحت وب است.
- تکامل پیوسته
  - برخلاف نرم افزارهای کاربردی سنتی، برنامه های تحت وب پیوسته در حال تکامل هستند.

ماهیت برنامه های کاربردی تحت وب

- بی واسطگی
  - نیاز اجباری برای رساندن سریع نرم افزار به بازار است.
- امنیت
  - برای محافظت از محتویات حساس معیارهای امنیتی قوی ای باید پیاده سازی شود.
- زیبایی شناسی
  - یک بخش غیر قابل انکار از جاذبه ی برنامه های تحت وب ظاهر آنهاست.
  - به اندازه موفقیت در طراحی فنی اهمیت دارد.

لایه های مهندسی نرم افزار

لایه های مهندسی نرم افزار

- مهندسی نرم افزار یک فن آوری لایه ای است که سنگ بنای آن توجه به کیفیت است.
- لایه ی فرآیند یک چارچوب را برای تحویل موثر فناوری وضع می کند.
- روشهای مهندسی نرم افزار، شیوه های فنی برای ساخت نرم افزار را فراهم می آورند.
- ابزارهای مهندسی نرم افزار متضمن پشتیبانی خودکار یا نیمه خودکار برای فرآیند و روشها هستند.

فرآیند نرم افزار

- فرایند مجموعه ای از فعالیت ها، کنش ها و وظایف است که هنگام ایجاد یک محصول کاری اجرا می شوند.
  - یک فعالیت، کوششی است در جهت رسیدن به هدفی گسترده (مانند برقراری ارتباط با افراد ذی نفع)
  - یک کنش شامل مجموعه ای از وظایف است که یک محصول کاری عمده را تولید می کند(مانند مدل طراحی معماری).
- در حیطه مهندسی نرم افزار فرایند یک روش انطباق پذیر است که تیم نرم افزار به کمک آن می توانند مجموعه ای مناسب از کنشها و وظایف کاری را برگزینند.

فرآیند نرم افزار

- یک چارچوب فرآیند کلی برای مهندسی نرم افزار شامل پنج فعالیت می شود:
- ارتباطات
  - هدف ارتباطات درک اهداف طرف های ذینفع برای پروژه و جمع آوری خواسته هایی است که می توانند ویژگی ها و قابلیتهای نرم افزار را تعیین کنند.
- برنامه ریزی
  - با توصیف وظایف فنی که قرار است اجرا شوند، خطرات احتمالی، منابعی که مورد نیاز خواهد بود، محصولات کاری که باید تولید شوند و زمان بندی کاری، مهندسی نرم افزار را مشخص می کند.

فرآیند نرم افزار

- مدل سازی
  - مهندسی نرم افزار جهت درک بهتر خواسته ها و طراحی که به این خواسته ها برسد این 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 شده که محصول تحویل داده شده را ارزیابی و براساس ارزیابی بازخوردی ارائه کند.

فعالیتهای چتری

- این فعالیت ها در سرتاسر فرایند نرم افزار رخ می دهد و کانون توجه آن ها اساسا مدیریت پروژه،پیگیری و کنترل است.
- ۱. کنترل و پیگیری پروژه های نرم افزاری
- ۲. مدیریت ریسک
- ۳. تضمین کیفیت نرم افزار
- ۴. بازبینی های فنی
- ۵. اندازه گیری
- ۶. مدیریت پیکربندی نرم افزار
- ۷. مدیریت قابلیت استفاده مجدد
- ۸. تهیه و تولید محصول کاری

تفاوت فرآیندها در پروژه های مختلف

- جریان کلی فعالیت ها، کنشها و وظایف و بستگی آن ها به یکدیگر
- درجه تعریف کنش ها و وظایف در هر فعالیت چارچوبی
- درجه شناسایی محصولات کاری و نیاز به آن ها
- شیوه اعمال فعالیت های تضمین کیفیت
- درجه کلی جزئیات به کار رفته در توصیف فرایند
- درجه دخالت مشتری و طرف های ذینفع در پروژه
- سطح استقلال داده شده به تیم نرم افزار
- درجه توصیف نقشها و سازماندهی تیم

مدل فرآیند تجویزی(فصل ۲)

- مدلهای فرایند تجویزی بر جزئیات تعریف، شناسایی و کاربرد فعالیت ها و وظایف تاکید دارند.
- هدف آن ها بهبود بخشیدن به کیفیت سیستم، بالا بردن قابلیت مدیریت پروژه، قابل پیش بینی کردن تاریخ تحویل و هزینه ها و راهنمایی تیم مهندسان نرم افزار برای کارهای لازم است.

مدل فرآیند چابک(فصل ۳)

- مدل فرایند چابک بر سرعت تاکید دارد و مجموعه ای از اصول را دنبال می کند که به یک روش غیر رسمی تر برای فرایند نرم افزار منجر می شود.
- این مدل بر قابلیت مانور و انطباق پذیری تاکید دارد و برای انواع بسیاری از پروژه ها به ویژه مهندسی برنامه کاربردی تحت وب مناسب است.

اصول کلی در مهندسی نرم افزار

- واژه ی اصل به معنای یک قانون یا فرض زیر بنایی است که در سیستم فکری وجود آن ضروری است.
- اصول کلی دیوید هوکر در مهندسی نرم افزار عبارتند از:
- ۱(دلیل وجود سیستم
  - هر سیستم به وجودی نیاز دارد که برای کاربرانش ارزشی فراهم کند
- ۲(ساده نگه داشتن
  - سادگی سبب ایجاد یک سیستم قابل فهم تر با قابلیت نگهداری بالاتر و کم خطا تر می شود.
- ۳(حفظ چشم انداز

اصول کلی در مهندسی نرم افزار

- ۴) آنچه که شما تولید می کنید دیگران مصرف کنند.
  - همواره طراحی و پیاده سازی را طوری انجام دهید که دیگران قادر به درک آن باشند.
- ۵(آینده نگری
  - سیستم با طول عمر بالا از ارزش بیشتری برخوردار است.
  - مسئله را در حالت کلی حل کنید نه در حالت خاص
- ۶(برنامه ریزی پیشاپیش برای استفاده مجدد
  - استفاده مجدد سبب صرفه جویی در کار و زمان می شود.
- ۷(تفکر
  - تفکر قبل از انجام هر کاری سبب نتایج بهتری خواهد شد.

پندارهای باطل نرم افزاری

پندارهای باطل مدیریتی

- پندار باطل : ما از قبل کتابی داریم که آکنده از استانداردها و روال های لازم برای ساختن نرم افزارهاست. آیا این کتاب آنچه را که افراد من باید بدانند در اختیارشان قرار نخواهد داد؟
- واقعیت : با وجود کتاب، آیا کامل است؟ آیا سازندگان از وجود آن آگاهند؟ آیا مهندسی نرم افزار نوین را ارائه می دهد؟ .... خیر

پندارهای باطل مدیریتی

- پندار باطل : اگر از برنامه عقب بیفتیم، می توانیم بر تعداد برنامه نویسان بیفزاییم و عقب ماندگی را جبران کنیم (یورش مغولی)
- واقعیت : با افزودن افراد در پروژه ای که تاخیر دارد بر تاخیر آن افزوده می شود.
- پندار باطل : اگر تصمیم به برون سپاری یک پروژه نرم افزاری به شرکت دیگری بگیرم می توانم خودم را آسوده سازم و بگذارم تا آن شرکت آن را بسازد.

پندارهای باطل مشتریان

- پندار باطل : بیانی کلی از اهداف، برای شروع به نوشتن برنامه ها کفایت می کند، جزئیات را بعدا می توانیم پر کنیم.
- واقعیت : بیان مبهم اهداف دستورالعملی برای مصیبت است.
- پندار باطل : نیازهای پروژه پیوسته در حال تغییر است، ولی این تغییرات را به راحتی می توان در نرم افزار جای داد،زیرا نرم افزار انعطاف پذیر است.
- واقعیت : تاثیر تغییر به زمان اعمال آن بستگی دارد. با گذر زمان هزینه ها به سرعت بالا رفته و تغییر مشکل می شود.

پندارهای باطل سازندگان

- پندار باطل : هنگامی که برنامه را نوشتیم و برنامه کار کرد، دیگر کار تمام است.
- واقعیت: هرچه زودتر دست به کار نوشتن دستورهای برنامه شوید، زمان بیشتری صرف به پایان بردن آن خواهید کرد. ۶۰ تا ۸۰ درصد کار روی نرم افزارها پس از نخستین بار تحویل آنها به مشتری صورت می پذیرد.
- پندار باطل : تا هنگامی که برنامه را اجرا نکرده ام، راهی برای ارزیابی کیفیت آن ندارم.
- واقعیت: مرور تکنیکی یک فیلتر کیفیتی است که از زمان آغاز پروژه قابل اجراست(فصل ۱۵)

پندارهای باطل سازندگان

- پندار باطل : تنها چیز قابل تحویل برای یک پروژه موفق، برنامه ای است که کار کند.
- واقعیت: انواع محصولات کاری از قبیل مدلها، مستندات، طرحها و برنامه کاری
- پندار باطل : مهندسی نرم افزار، ما را وادار می سازد که مستندات حجیم و بیهوده تهیه کنیم و از سرعت ما می کاهد.
- واقعیت: مهندسی نرم افزار مستند سازی نیست بلکه به ایجاد محصول با کیفیت بهتر می انجامد. کیفیت بهتر سبب دوباره کاری کمتر و دوباره کاری کمتر سبب تحویل سریعتر می شود.