

دانشگاه فنی و حرفه ای

دانشکده فنی و حرفه ای امام جعفر صادق (ع) آستانه اشرفیه

فهرست

SOFTWARE AND SOFTWARE ENGINEERING

۱- نرم افزار و مهندسی نرم افزار

PROCESS MODELS

۲- مدل های فرایند

AGILE DEVELOPMENT

۳- توسعه ی چابک (سریع)

PRINCIPLES THAT GUIDE PRACTICE

۴- اصول راهنما در مهندسی نرم افزار

UNDERSTANDING REQUIREMENTS

۵- شناخت خواسته ها

REQUIREMENTS MODELING

۶- مدل سازی خواسته ها: سناریوها

REQUIREMENTS MODELING : FLOW, BEH AVIOR

۷- مدل سازی خواسته ها : جریان ، رفتار

DESIGN CONCEPTS

۸- مفاهیم طراحی

فصل اول :

نرم افزار و مهندسی نرم افزار

The Nature of Software

Defining Software

Software Application Domains

Legacy Software

The Unique Nature of WebApps

Software Engineering

The Software Process

Software Engineering Practice

The Essence of Practice

General Principles

Software Myths

How It All Starts



نرم افزار چیست؟

محصولی است که مهندس نرم افزار طراحی می کند و می سازد.

شامل برنامه هایی است که در کامپیوتری به هر اندازه و با هر معماری قابل اجرا هستند.

مستنداتی دارد که شامل فرم های واقعی و مجازی می شود.

داده هایی دارد که ترکیبی از ارقام و حروف است و می تواند شامل اشکال نمایشی از قبیل اطلاعات تصویری، صوتی و ویدئویی باشد.

محصول کار چیست؟

از دیدگاه مهندس نرم افزار: برنامه ها، مستندات و داده ها که نرم افزار کامپیوتری است.

از دیدگاه کاربر: اطلاعاتی است که به نحوی به درد کاربر می خورد.

آیا نرم افزار مرده است؟

اگر چنین بود ، این کتاب را نمی خواندید.

ماهیت نرم افزار

THE NATURE OF SOFTWARE

امروزه نرم افزار نقشی دو گانه دارد:

۱- نوعی محصول است.

توان بالقوه ی یک سخت افزار یا شبکه ای از کامپیوترها را بالفعل می کند.

۲- وسیله ی نقلیه ای برای تحویل یک محصول.

مبنای کنترل کامپیوتر، مخابرات اطلاعات و خلق و کنترل برنامه های دیگر را تشکیل می دهد.

نرم افزار چیست؟

۱- دستورالعمل ها (instructions)

۲- ساختمان داده ها (data structures)

۳- اطلاعات توصیفی (documentation)

Software

Hardware



خصوصیات نرم افزار

(۱) نرم افزار، مهندسی و بسط داده می شود و چیزی نیست که به معنای کلاسیک کلمه ساخته شود.

(۲) نرم افزار فرسوده نمی شود.

(۳) گرچه صنعت در حال حرکت به سوی مونتاژ قطعات است ولی اکثر نرم افزارها همچنان به

صورت سفارشی ساخته می شوند.

نمودار آهنگ شکست سخت افزار

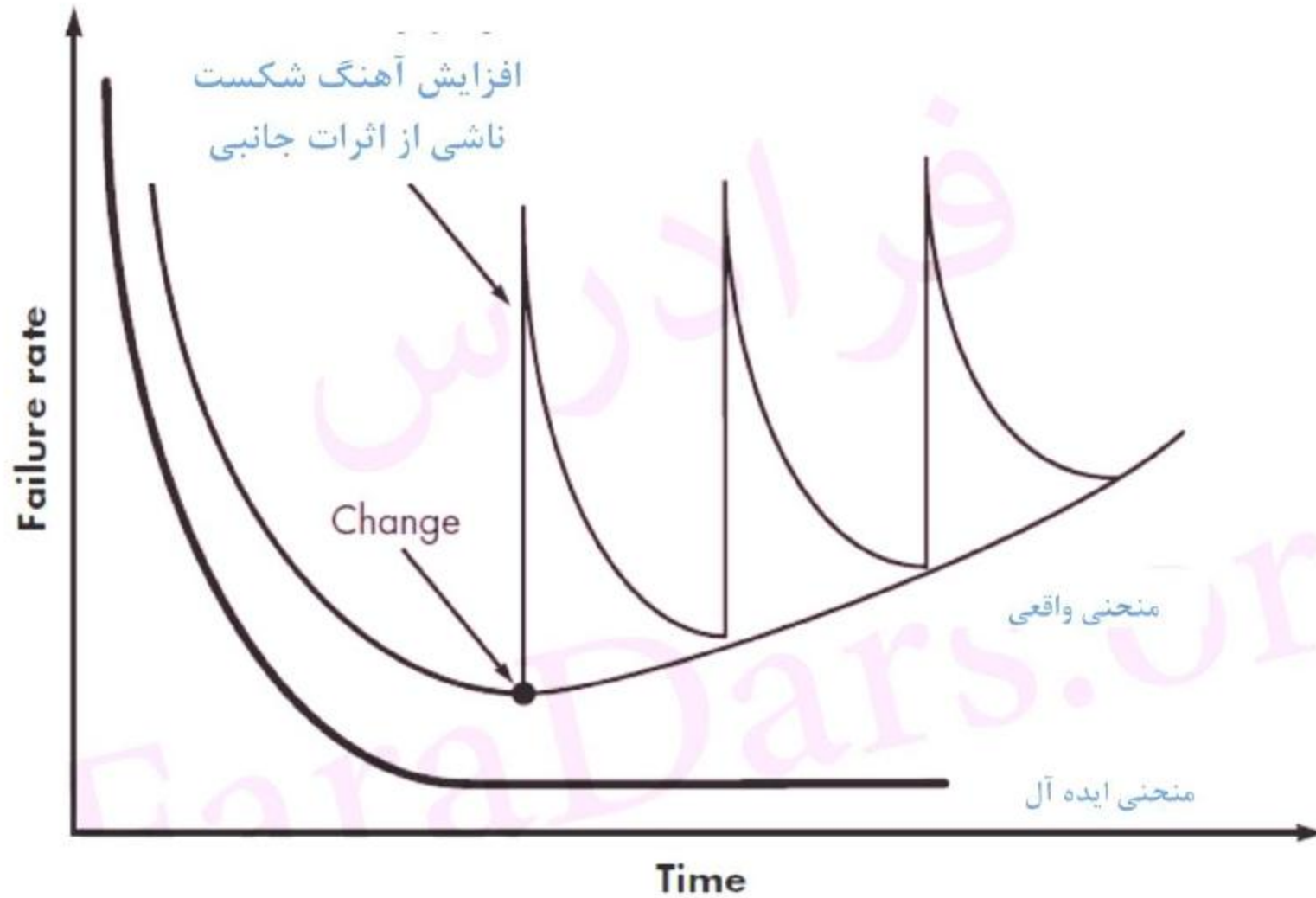


این نمودار نشان می دهد که سخت افزار در ابتدای عمر آهنگ شکست شدیدی دارد.

این عیوب تصحیح می شوند و آهنگ شکست در یک دوره زمانی به مقدار ثابتی نزول می کند.

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

منحنی های شکست واقعی و ایده آل برای نرم افزار



نرم افزار در دوران حیات خود دستخوش تغییر می شود با اعمال این تغییرات احتمال دارد برخی عیوب جدید وارد شوند و باعث خیز منحنی آهنگ شکست می شود و پیش از آنکه منحنی بتواند به آهنگ شکست منظم اولیه خود برسد تغییر دیگری درخواست می شود که باعث خیز دوباره منحنی می شود.

دامنه های کاربرد نرم افزار

(۱) سیستمی : system software

(۲) کاربردی : application software

(۳) مهندسی/علمی : engineering/scientific software

(۴) تعبیه شده : embedded software

(۵) خط تولید : product-line software

(۶) برنامه های کاربردی تحت وب: WebApps

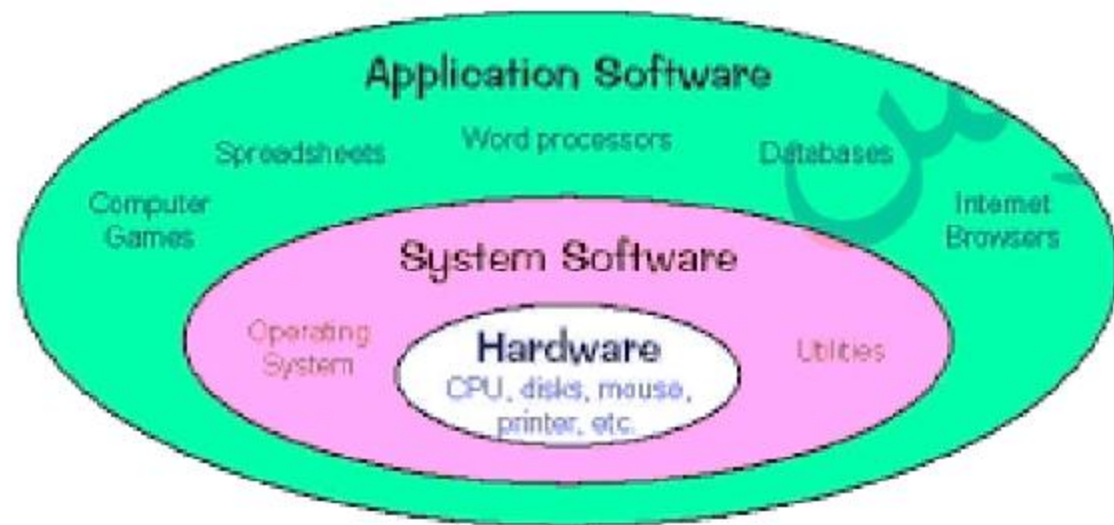
(۷) هوش مصنوعی: AI software

دامنه های کاربرد نرم افزار

(۱) نرم افزار های سیستمی : system software

مجموعه ای از برنامه ها که برای سرویس دهی به برنامه های دیگر نوشته شده اند.

مشخصه های حیطه ی نرم افزار سیستمی عبارتند از:



الف) تعامل سنگین با سخت افزار

ب) استفاده سنگین توسط چند کاربر

ج) عمل کنونی که مستلزم زمانبندی است

د) مدیریت فرایند پیچیده و اشتراک منابع

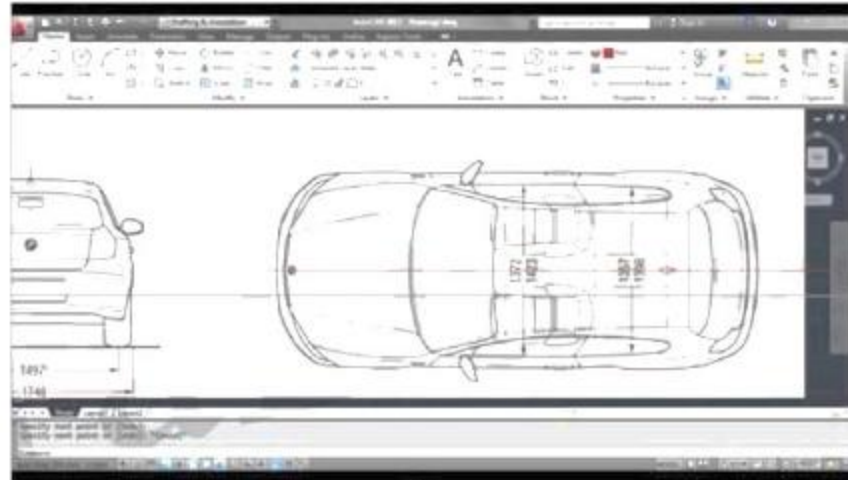
ه) ساختمان داده های پیچیده و واسط های خارجی دیگر

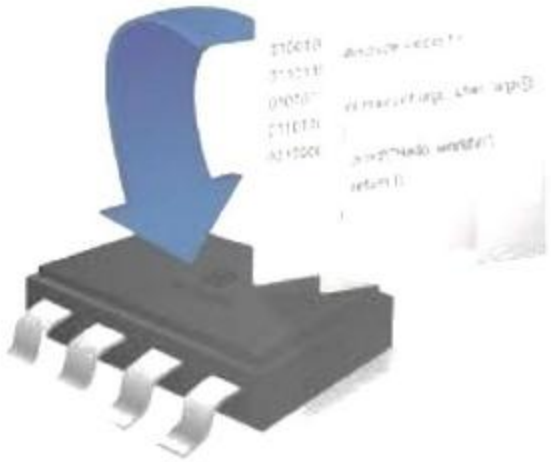
(۲) نرم افزارهای کاربردی : application software

برنامه های مستقل که یک نیاز تجاری را برطرف می سازد.

۳) نرم افزارهای مهندسی/علمی : engineering/scientific software

نرم افزارهای علمی توسط الگوریتم هایی مشخص می شوند که ارقام و اعداد را پردازش می کنند.





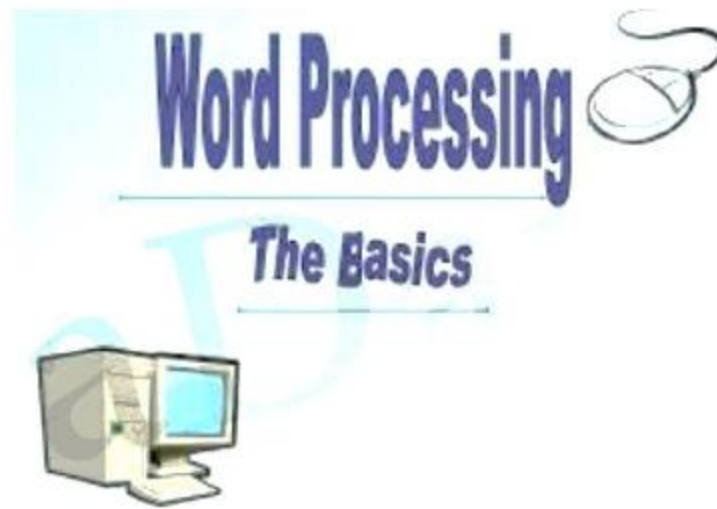
۴) نرم افزارهای تعبیه شده : embedded software

در حافظه فقط خواندنی جای دارند.

برای کنترل محصولات و سیستم های مربوط به بازارهای صنعتی و مصرفی به کار می روند.

۵) نرم افزارهای خط تولید : product-line software

برای فراهم آوردن یک قابلیت خاص جهت استفاده توسط بسیاری از مشتریان مختلف طراحی می شوند.





۶) برنامه های کاربردی تحت وب: WebApps

این گروه از نرم افزارهای شبکه ای شامل مجموعه گسترده ای از برنامه های کاربردی می شوند.



۷) نرم افزار های هوش مصنوعی: AI software

برای حل مسائل پیچیده ای که به روش های عددی قابل حل نیستند از الگوریتم های غیر عددی استفاده می کنند.

مثال هایی از کاربرد :

- سیستم های خبره
- تشخیص الگو های تصویری و صوتی
- شبکه های عصبی مصنوعی
- اثبات قضایا و بازی

چالش ها

- کار با کامپیوتر در جهانی باز (Open world computing)
- تامین منابع از طریق شبکه (Netsourcing)
- کد منبع باز (Open source)

نرم افزارهای قدیمی

نرم افزارهای قدیمی چند دهه قبل ساخته شده اند و پیوسته اصلاح شده اند تا تغییرات به عمل آمده در خواسته های تجاری را پاسخ گو باشند.

دلایل تکامل سیستم های قدیمی:

- ۱) نرم افزار باید برای برآورده ساختن نیازهای محیط های جدید کامپیوتر یا فن آوری های جدید اصلاح گردد.
- ۲) نرم افزار باید بهبود یابد تا خواسته های تجاری جدید را پیاده سازی کند.
- ۳) نرم افزار باید گسترش داده شود تا با سایر سیستم ها یا بانک اطلاعاتی جدیدتر قابلیت همکاری داشته باشد.
- ۴) نرم افزار باید دوباره معماری شود تا در یک محیط شبکه نیز قادر به حیات خود باشد.

مشخصات برنامه های تحت وب

Characteristics of WebApps

- ۱- میزان تمرکز شبکه (**Network intensiveness**)
- ۲- همروندی (**Concurrency**)
- ۳- بار غیر قابل پیش بینی (**Unpredictable load**)
- ۴- کارایی (**Performance**)
- ۵- قابلیت دسترسی (**Availability**)
- ۶- داده - محوری (**Data driven**)
- ۷- حساس به محتویات (**Content sensitive**)
- ۸- تکامل پیوسته (**Continuous evolution**)
- ۹- بی واسطگی (**Immediacy**)
- ۱۰- امنیت (**Security**)
- ۱۱- زیبایی شناسی (**Aesthetics**)



چند نکته کلیدی در مورد ساخت نرم افزار

- (۱) پیش از آنکه برای مساله راهکاری بیابید آن را درک کنید.
- (۲) طراحی ، یکی از فعالیت های محوری در مهندسی نرم افزار است.
- (۳) کیفیت و قابلیت نگهداری هر دو نتیجه طراحی خوب هستند.

تعریف مهندسی نرم افزار

تعریف فریتز باور:

وضع اصول مهندسی بجا و مناسب و استفاده از آنها برای به دست آوردن یک نرم افزار مقرون به صرفه که قابل اطمینان بوده، روی ماشین های واقعی به طرز کارآمد عمل کند.

تعریف IEEE

- ۱- کاربرد یک روش سیستماتیک ، علمی و کمیت پذیر در بسط، راه اندازی و نگهداری نرم افزار ، یعنی استفاده از مهندسی در نرم افزار .
- ۲- مطالعه روش ها به صورت ذکر شده در (۱)

لایه های مهندسی نرم افزار



- توجه به کیفیت، سنگ بنای نگهدارنده ی مهندسی نرم افزار است.
- بنیاد مهندسی نرم افزار لایه ی فرایند است.
- روش های مهندسی نرم افزار، شیوه های فنی برای ساخت نرم افزار را فراهم می کند.
- ابزار مهندسی نرم افزار متضمن پشتیبانی خودکار یا نیمه خودکار برای فرایند و روش ها هستند.

تعریف فرایند

فرایند تعیین می کند که چه کسی چه کاری را در چه زمانی و چگونه انجام دهد تا به هدفی معین برسد.

چار چوب فرایند کلی برای مهندسی نرم افزار شامل ۵ فعالیت می شود

۱- ارتباطات (Communication)

هدف ارتباطات درک اهداف طرف های ذینفع برای پروژه و جمع آوری خواسته هایی است که می توانند ویژگی ها و قابلیت های عملیاتی نرم افزار را تعیین کنند.

۲- برنامه ریزی (Planning)

با توصیف وظایف فنی که قرار است اجرا شوند، خطرات احتمالی، منابعی که مورد نیاز خواهند بود، محصولات که باید تولید شوند و زمانبندی کاری، مهندسی نرم افزار را مشخص می کند.

۳- مدل سازی (Modeling)

مهندس نرم افزار با ایجاد مدل هایی جهت درک بهتر خواسته ها و طراحی که به این خواسته ها برسد مدل سازی می کند.

۴- ساخت (Construction)

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

۵- استقرار (Deployment)

نرم افزار به مشتری تحویل می شود که محصول تحویل شده را ارزیابی و براساس آن بازخوردی ارائه می دهد.

فعالیت های چتری (Umbrella Activities)

- مدیریت پروژه های نرم افزاری (Software project management)
- مدیریت ریسک (Risk management)
- تضمین کیفیت نرم افزار (Software quality assurance)
- بازبینی های فنی (Formal technical reviews)
- اندازه گیری (Measurement)
- مدیریت پیکر بندی نرم افزار (Software configuration management)
- مدیریت قابلیت استفاده مجدد (Reusability management)
- تهیه و تولید محصول کاری (Work product preparation and production)

تفاوت های مدل فرایند

فرایندی که برای یک پروژه پذیرفته می شود، ممکن است با فرایند پذیرفته شده برای پروژه ای دیگر تفاوتی چشمگیر داشته باشد. از جمله :

(۱) جریان کلی فعالیت ها ، کنش ها و وظایف و بستگی آن ها به یکدیگر

(۲) درجه تعریف کنش ها و وظایف در هر فعالیت چارچوبی

(۳) درجه ی شناسایی محصولات کاری و نیاز به آن ها

(۴) شیوه ی اعمال فعالیت های تضمین کیفیت

(۵) درجه ی کلی جزئیات به کار رفته در توصیف فرایند

(۶) درجه ی دخالت مشتری و طرف های ذینفع در پروژه

(۷) سطح استقلال داده شده به تیم نرم افزار

(۸) درجه ی توصیف نقش ها و سازمان دهی تیم

مدل فرایند تجویزی

تاکید بر جزئیات تعریف، شناسایی و کاربرد فعالیت ها و وظایف .

هدف :

- بهبود بخشیدن به کیفیت سیستم،
- بالا بردن قابلیت مدیریت پروژه ها،
- قابل پیش بینی کردن تاریخ تحویل و هزینه ها
- راهنمایی تیم مهندسان نرم افزار در اجرای کارهای لازم برای ساخت یک سیستم.

مدل فرایند چابک

بر سرعت تاکید دارد و مجموعه ای از اصول را دنبال می کند که به یک روش غیررسمی تر برای فرایند نرم افزار منجر می شود.

این مدل بر قابلیت مانور و انطباق پذیر تاکید دارد.

برای انواع بسیاری از پروژه ها مناسب بوده، به بویژه هنگام مهندسی برنامه کاربردی تحت وب.

جوهر عمل در مهندسی نرم افزار

۱- شناخت مسئله (برقراری ارتباط و تحلیل)

Understand the problem (communication and analysis)

۲- طرح ریزی برای یک راه حل (مدل سازی و طراحی نرم افزار)

Plan a solution (modeling and software design)

۳- اجرای برنامه ریزی (ایجاد کد)

Carry out the plan (code generation)

۴- بررسی نتیجه برای صحت (آزمایش و تضمین کیفیت)

Examine the result for accuracy (testing and quality assurance)

اصول کلی در مهندسی نرم افزار

Hooker's General Principles

اصل : یک قانون یا فرض زیر بنایی است که در یک سیستم فکری وجود آن ضروری است.

۱- دلیل وجود سیستم (The Reason It All Exists)

۲- ساده نگه داشتن (Keep It Simple)

۳- حفظ چشم انداز (Maintain the Vision)

۴- آنچه که شما تولید می کنید، دیگران مصرف کنند (What You Produce, Others Will Consume)

۵- آینده نگری (Be Open to the Future)

۶- برنامه ریزی پیشاپیش برای استفاده مجدد (Plan Ahead for Reuse)

۷- تفکر (Think!)

پندارهای باطل نرم افزار

Software Myths

(۱) پندارهای باطل مدیریتی

(۲) پندارهای باطل مشتریان

(۳) پندارهای باطل سازندگان

پندارهای باطل مدیریتی

اگر از برنامه عقب بیفتیم ، می توانیم بر تعداد برنامه نویسان بیفزاییم و عقب افتادگی را جبران کنیم.

اگر تصمیم به برون سپاری (**outsource**) یک پروژه ی نرم افزاری به شرکت دیگری بگیرم ، می توانم خودم را آسوده سازم و بگذارم تا آن شرکت آن را بسازد.

پندهای باطل مشتریان

Customer myths

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.

پندارهای باطل سازندگان

تا هنگامی که برنامه را اجرا نکرده ایم، راهی برای ارزیابی کیفیت آن ندارم.

تنها چیز قابل تحویل برای یک پروژه ی موفق، برنامه ای است که کار می کند.

مهندسی نرم افزار ما را وادار می کند که مستندات حجیم و بیهوده تهیه کنیم و از سرعت کار ما می کاهد.

چگونگی آغاز یک پروژه

SAFEHOME



How a Project Starts

The scene: Meeting room at CPI Corporation, a (fictional) company that makes consumer products for home and commercial use.

The players: Mal Golden, senior manager, product development; Lisa Perez, marketing manager; Lee Warren, engineering manager; Joe Camalleri, executive VP, business development

مدیر بازرگانی

مدیر ارشد



معاون مدیر اجرایی

مدیر مهندسی

پایان فصل اول