با نام خدا و سلام، در این مجموعه بحث که در پیش خواهیم داشت قصد دارم به موضوع مهندسی نرم افزار بپردازم.

اهداف در قسمت اول مجموعه مباحث مهندسی نرم افزار:

- آشنایی با مفهوم «مهندسی نرم افزار» و چیستی آن و چرا اهمیت دارد.
- فهم و درک اینکه توسعه انواع مختلف سیستم های نرم افزاری نیاز به تکنیک های مختلف مهندسی نرم افزار دارند.
- آشنایی با مسائل حرفه ای و اصولی برای مهندسی نرم افزار

مرجع اصلی در مجموعه مطالبی که در حوزه «مهندسی نرم افزار» در خدمتتون هستم کتاب "Software Engineering" نوشته ی "Ian Summervil" و ویرایش دهم آن می باشد. مطالب پیش رو به هیچ عنوان کپی شده نیست 😊 ممنون میشم اگر از مطالب استفاده ای می کنید، مرجع مطالب (سایت صادق خان یا محمد صادق صالحی) را قید کنید. خوشحال میشم به وبلاگم سر بزنید :

http;//sadegh-khan.ir

خلاصه فصل اول کتاب سامرویل

در سال ۲۰۱۶ بیش از ۷۵ درصد از جمعیت جهان دارای گوشی های هوشمند و در نتیجه موبایل های کنترل شده توسط نرم افزار داشتند و تمامی آنها امکان دسترسی به اینترنت را نیز داشتند.

سیستم های نرم افزاری انتزاعی و ناملموس می باشند. محدودیت های همچون ویژگی های مادی و قوانین فیزیکی ندارند. همین قضیه مهندسی نرم افزار را ساده می کند زیرا که هیچ محدودیت طبیعی برای آن وجود ندارد.اگرچه به دلیل وجود نداشتن محدودیت های فیزیکی، سیستم های نرم افزاری می توانند به سرعت بسیار پیچیده شوند، درکشان سخت شود، و در نتیجه تغییرات در آنها هزینه بردار می شود.

انواع مختلف سیستم های نرم افزاری وجود دارد، از سیستم های تعبیه شده ساده تا سیستم های اطلاعاتی پیچیده جهانی. نکته اینجاست که هیچ متُد و یا تکنیک جامعی برای مهندسی نرم افزار وجود ندارد، چرا؟، به دلیل آنکه انواع مختلف نرم افزار ها نیازمند رویکردهای متفاوت می باشد. در نتیجه همه سیستم های نرم افزاری نیاز به مهندسی نرم افزاری دارند، اما همه شان تکنیک یا متُد مهندسی نرم افزار یکسان ندارند.

در حال حاضر «خرابی نرم افزار» زیاد گزارش می شود و مهندسی نرم افزار را برای توسعه نرم افزارهای مدرن، ناکافی دانسته و آن را نکوهش می کنند. اگرچه به نظر میرسه اکثر این خرابی ها حاصل دو فاکتور زیر باشند:

۱. افزایش پیچیدگی سیستم، همین طور که تکنیک های جدید مهندسی نرم افزار به ما کمک در ساخت سیستم های پیچیده گرده است، مطالبات را نیز تغییر داده است. سیستم ها باید سریعتر ساخته و آماده تحویل شوند، در این صورت سیستم های بزرگتر و حتی پیچیده تر نیاز می شود و سیستم ها باید دارای قابلیت های جدیدی باشند که در گذشته پنداشت می شد که امکان پذیر نیستند. برای مقابله با چالشهای جدید آماده سازی نرم‌افزار های پیچیده تر، باید تکنیک های مهندسی نرم افزار جدیدی آماده شوند.
۲. خرابی در استفاده از متدهای مهندسی نرم‌افزار، به وضوح نوشتن برنامه کامپیوتر بدون استفاده از متد یا تکنیک مهندسی نرم افزار ساده می باشد. شرکت ها به دلیل تحول در خدمات و محصولات شان، به سمت توسعه نرم افزار روی آورده اند. آنها در کارهای روزانه شان از متدهای مهندسی نرم افزار استفاده نمی‌کنند. در نتیجه نرم‌افزارشان اغلب هزینه بر و با قابلیت اعتماد کمتر از چیزی که انتظارش را داشته اند، می شود. برای رفع این مشکل ما به آموزش و یادگیری مهندسی نرم افزار بهتری نیاز داریم.

تاریخچه مهندسی نرم افزار:

باور به مهندسی نرم افزار ابتدا در سال ۱۹٦۸ در کنفرانسی مطرح شد که بعدا «نقطه عطف نرم‌افزار» نامیده شد. در این کنفرانس مشخص شد که رویکردهای فردی برای توسعه برنامه، برای سیستم های نرم‌افزاری پیچیده ی بزرگ قیاس پذیر نمی باشد. غیر قابل اعتماد بوده و بیش از حد تعیین شده هزینه بردار بوده و به زمان تحویل تعیین شده نمی رسیدند و معمولا با تاخیر به مشتری تحویل داده می شدند.

بین سال های دهه ۷۰ و ۸۰ میلادی، تنوعی از تکنیک ها و متدهای مهندسی نرم افزار توسعه داده شدند، به عنوان نمونه : برنامه نویسی ساختاریافته و توسعه شیئ محور. ابزارهای استاندارد در آن سال‌ها توسعه داده شدند که شالوده مهندسی نرم افزار امروزه را ساخته‌اند.

توسعه نرم افزار حرفه ای:

مهندسی نرم افزار قصد دارد که توسعه نرم افزار حرفه ای را پشتیبانی کنه، نه برنامه نویسی شخصی را. که شامل تکنیک هایی است که از مشخصات برنامه، طراحی و تکامل برنامه پشتیبانی می کند و نکته اینجاست که هیچ کدام این موارد به طور معمول با توسعه برنامه شخصی ارتباطی ندارند. برای کمک به اینکه دید گسترده بر مهندسی نرم افزار داشته باشید، به جدول۱-۱ توجه داشته باشید.

پرسش	پاسخ
نرم افزار چیست؟	برنامه های کامپیوتری و مستندات مرتبط با آن. محصولات نرم افزاری ممکن است برای یک مشتری مشخص یا برای یک بازار عام طراحی و توسعه داده شوند.
ویژگی های یک نرم‌افزار خوب چیست؟	نرم افزار خوب باید عملکرد و کارایی لازم را در اختیار کاربر قرار دهد و باید قابل نگهداری، قابل اطمینان و قابلیت استفاده داشته باشد.
مهندسی نرم‌افزار چیست؟	مهندسی نرم‌افزار یک انتظام مهندسی می باشد که نگرانی اش کلیه جنبه های تولید نرم افزار از مفاهیم اولیه تا بهره‌برداری و نگهداری می باشد.
فعالیت های اساسی مهندسی نرم‌افزار چیست؟	مشخصات نرم افزار، توسعه نرم‌افزار، اعتبار سنجی نرم‌افزار و تکامل نرم افزار.
تفاوت بین مهندسی نرم‌افزار و علوم کامپیوتر چیست؟	علوم کامپیوتر تمرکز روی تئوری و مفاهیم پایه دارد، مهندسی نرم‌افزار توجه‌اش عملی بودن توسعه و ارائه نرم افزار مفید می باشد.
تفاوت بین مهندسی نرم‌افزار و مهندسی سیستم چیست؟	مهندسی سیستم توجه بر تمامی جنبه های توسعه سیستم های کامپیوتری شامل سخت‌افزار، نرم‌افزار و مهندسی فرآیند ها دارد. مهندسی نرم‌افزار قسمتی از این فرآیند عمومی تر می باشد.
چالش های اساسی مهندسی نرم‌افزار چیست؟	مقابله با افزایش تنوع، تقاضا برای کاهش زمان تحویل و توسعه نرم افزاری قابل اعتماد
هزینه های مهندسی نرم‌افزار چیست؟	تقریباً ٦٠ درصد از هزینه نرم‌افزار هزینه های توسعه آن می باشد و ٤٠ درصد هزینه تست کردن آن. برای نرم‌افزارهای سفارشی، هزینه های تکامل عموما بیشتر از هزینه های توسعه می باشد.

بهترین متد و تکنیک‌های مهندسی نرم‌افزار چیست؟	به این دلیل که همه ی پروژه های نرم‌افزاری باید به طور حرفه ای مدیریت و توسعه شوند، تکنیک های متفاوتی برای انواع مختلف سیسستم ها مناسب می باشد. برای مثال، بازی‌ها باید همیشه با مجموعه ای از پروتوتایپ ها توسعه داده شوند، درحالیکه سیستم های کنترل ایمنی بحرانی، نیازمند مشخصات کامل و قابل تحلیلی دارند تا توسعه داده شوند. متد یا روشی وجود ندارد که برای همه چیز خوب باشد.
چه تفاوت‌هایی را اینترنت در مهندسی نرم‌افزار ایجاد کرده است؟	نه تنها اینترنت منجر به توسعه سیستم های بسیارگسترده، بزرگ و مبتنی بر سرویس شده است، همچنین منجر به پشتیبانی از ساخت اپلیکیشن برای دستگاه های موبایل شده است که خود باعث تغییر در اقتصاد نرم افزار شده است.

خیلی از افراد فکر می کنند نرم افزار (Software) به سادگی و به لغتی دیگر همان برنامه های کامپیوتری (Computer Program) می باشد. در حالیکه وقتی ما از مهندسی نرم افزار صحبت می کنیم، نرم افزار فقط خود برنامه ها نمی باشد، بلکه تمامی مستندات مرتبط، کتابخانه ها، وب سایت های پشتیبانی، و داده های پیکر بندی مورد نیاز برای بهبود این برنامه ها را شامل می شود. یک سیستم نرم افزار حرفه ای توسعه یافته، اغلب بیشتر از یک برنامه می باشد. یک سیستم ممکن است شامل چندین برنامه جدا باشد که از فایل های پیکربندی جهت تنظیم همین برنامه ها استفاده می کند

این یکی از مهمترین تفاوت های بین توسعه نرم افزار حرفه ای و آماتور می باشد زمانی که شما برنامه ای برای خودتان می نویسید، نگران این نیستید که سندی برای روش استفاده آن بسازید، چرا که فقط خود شما از آن استفاده خواهید کرد. در حالیکه اگر نرم افزاری توسعه می دهید که افراد دیگر از آن استفاده می کنند و مهندسین دیگری آن را تغییر خواهند داد، شما باید اطلاعات اضافی دیگری در کنار کُد فراهم کنید.

مهندسین نرم افزار، دغدغه شان در مورد نرم افزاری است که قرار به فروش آن به مشتری می باشد، دو نوع محصول نرم افزاری وجود دارد:

۱. محصولات عمومی (Generic products): این سیستم ها مستقل (stand-alone) هستند که توسط یک سازمان توسعه دهنده تولید شده‌اند و در بازار آزاد به هر مشتری که قادر به خرید آن باشد، فروخته می‌شوند. مثال برای این نوع محصولات میتوان: محصولاتی شامل اپ های موبایل، نرم‌افزار برای Pcها مانند پایگاه‌داده ها، ویرایشگر های متن، پکیج های طراحی و ابزار های مدیریت پروژه اشاره کرد.
۲. نرم‌افزار سفارشی (Customized (or bespoke) software): اینها سیستم‌هایی هستند که توسط مشتری خاصی و برای همان مشتری توسعه داده شده اند. پیمانکار نرم افزاری را برای مشتری و مختص آن طراحی و پیاده سازی می کند. مثال بر این نوع نرم‌افزارها: سیستم های کنترلی برای دستگاه‌های الکترونیکی، سیستم هایی که برای پشتیبانی فرآیند تجاری مشخصی نوشته می‌شوند و سیستم های کنترل ترافیک هوایی.

تمایز مهم بین این سیستم ها این هست که در محصولات عمومی، سازمانی که محصول را توسعه داده است، مشخصات نرم‌افزار را نیز کنترل می کند. به این معنی که اگر به مشکلات توسعه برخورد کنند، آنها می توانند مجدد فکر کنند که چه چیزی درحال توسعه است. در محصولات سفارشی، مشخصات توسط سازمانی که نرم‌افزار را خریداری کرده است، کنترل و توسعه داده می شوند. توسعه دهندگان نرم‌افزار باید با همان مشخصات کار کنند.

با این حال باز تمایز بین این دو دسته تقریبا مات و غیر شفاف است. در حال حاضر سیستم های زیادی ابتدا به صورت محصول عمومی توسعه داده می شوند و سپس نیازمندی های مشتری وفق داده می شوند. برا مثال سیستم های ERP.

مجموعه مشخصه های خاصی که شما ممکن است از سیستم های نرم‌افزاری انتظار داشته باشید به کاربرد آن سیستم بستگی دارد. در نتیجه سیستم کنترل هوایی باید ایمن باشد، یک بازی تعاملی باید واکنشگرا باشد و یک سیستم سوییچینگ تلفن نیز باید قابل اعتماد باشد و ... .اینها را میتوان به مجموعه ویژگی هایی نشان داد. در تصویر ۱.۲ ویژگی های اساسی یک سیستم نرم افزاری حرفه ای نشان داده شده است.

ویژگی محصول	توضیحات

مقبولیت (Acceptability)	نرم افزار باید برای کاربرانی که برای آنها طراحی شده، قابل پذیرش باشد. به این معنی که باید قابل فهم باشد، قابل استفاده و سازگار با دیگر سیستم هایی که آنها استفاده می کنند، باشد.
قابلیت اطمینان و امنیت (Dependability and security)	قابلیت اطمینان نرم افزار شامل گستره ای از ویژگی ها از جمله قابلیت اعتماد، امنیت و ایمنی می باشد. نرم افزار قابل اعتماد نباید بهنگام رخداد خطای سیستمی، منجر به آسیب های فیزیکی یا اقتصادی شود. نرم افزار باید امن باشد و در نتیجه کاربران مخرب نتوانند به سیستم دسترسی داشته باشند یا به آن آسیب وارد کنند.
کارایی (Efficiency)	نرم افزار نباید در استفاده از منابع سیستمی همچون مموری و پردازنده، ولخرجی داشته باشد.
قابلیت نگهداری (Maintainability)	نرم افزار باید به روشی نوشته شود که بتواند منجر به تغییرات مورد نیاز کاربران شود. این ویژگی بحرانی می باشد، چرا که تغییر نرم افزار نیاز غیر قابل امتناع از تغییر محیط بیزینس مورد نظر می باشد و باید بتوان با محیط جدید و تغییرات وفق پیدا کند.

http://sadegh-khan.ir/%d9%85%d9%87%d9%86%d8%af%d8%b3%db%8c-%d9%86%d8%b1%d9%85%e2%80%8c%d8%a7%d9%81%d8%b2%d8%a7%d8%b1-%d8%aa%da%a9%d9%86%db%8c%da%a9-%d9%87%d8%a7-%d9%88-%d8%b1%d9%88%d8%b4-%d9%87%d8%a7-%d9%85%d9%82%d8%af/

مهندسی نرم افزار:

مهندسی نرم افزار اصول مهندسی می باشد که دغدعه اش تمامی جنبه های تولید نرم افزار از مراحل اولیه مشخصات سیستم تا نگهداری از سیستم پس از استفاده از آن می باشد. در این تعریف «تمامی جنبه های تولید نرم افزار» نشان می دهد که مهندسی نرم افزار دغدغه اش تنها فرآیندهای تکنیکال نرم افزار نیست بلکه شامل تمامی فعالیت ها مانند مدیریت پروژه نرم افزار، توسعه و ابزارها نیز می باشد.

دو دلیل اهمیت مهندسی نرم افزار:

۱. هر روز بیشتر از دیروز جوامع و افراد متکی بر سیستم های نرم افزاری پیشرفته می شوند. ما نیاز داریم که سیستم های قابل اعتماد و اطمینان را به صورت اقتصادی و سریعتر بسازیم.
۲. در طولانی مدت، معمولا استفاده از متد های مهندسی نرم افزار برای سیستم های نرم افزاری پیشرفته، بجای نوشتن برنامه مطابق با روش های کدنویسی شخصی، مقرون به صرفه تر می باشد. عدم استفاده از روش مهندسی نرم افزار منجر به هزینه های بالاتر برای تست، تضمین کیفیت و نگهداری برای طولانی مدت می شود.

رویکرد سیستماتیکی که در مهندسی نرم افزار استفاده می شود را بعضی محافل یک فرآیند نرم افزار می نامند. یک فرآیند نرم افزار، توالی از فعالیت ها می باشد که منجر به تولید یک محصول نرم افزاری می شود. چهار فعالیت اساسی برای کلیه فرآیند های نرم افزاری به طور مشترک وجود دارد:

۱. مشخصات نرم افزار (Software specification): در اینجا مشتریان و مهندسین، نرم افزاری که تولید می شود و محدودیت ها و عملیات آن را تعریف می کنند.
۲. توسعه نرم افزار (Software development): جایی که نرم افزار طراحی و برنامه نویسی می شود.
۳. اعتبار سنجی نرم افزار (Software validation): در اینجا نرم افزار بررسی می شود که همان چیزی باشد که مشتری به آن نیاز داشته است.
۴. تکامل نرم افزار (Software evolution): اینجا نرم افزار برای تغییرات مشتری و نیازمندی های بازار اصلاح می شود.

انواع مختلف نرم افزار نیاز به فرآیندهای توسعه متفاوت دارند. برای مثال نرم افزار های برخط (real-time) در یک هواپیما باید قبل از شروع به توسعه به طور کامل مشخص باشند. در یک فروشگاه آنلاین معمولا مشخصات و برنامه نویسی با یکدیگر پیش می روند

همان طور که بعدا بیشتر در موردش توضیح خواهیم داد، انواع مختلفی از نرم افزار وجود دارد و در نتیجه یک روش یا تکنیک جامع برای مهندسی نرم افزار نداریم که از آن برای همه نرم افزار ها استفاده کنیم. چهار موضوع متفاوت روی انواع مختلف نرم افزار تاثیر گذار می باشد.

۱. ناهمگونی
۲. تغییرات اجتماعی و تجارت
۳. امنیت و اعتماد
۴. مقیاس

تنوع مهندسی نرم افزار

شاید مهمترین عامل در تعیین اینکه کدام روشها و تکنیک های مهندسی نرم افزار مهمتر هستند، نوع اپلیکیشنی هست که در حال توسعه می باشد. تنوع مختلفی از نرم افزار ها وجود دارد که شامل موارد زیر می شوند:

۱. اپلیکیشن های مستقل (Stand-alone applications) : این اپلیکیشن ها روی کامپیوترهای شخصی اجرا می شوند یا اپلیکیشن هایی که روی موبایل‌ها اجرا می شوند. تمامی عملکرهای لازم را دارند و نیازی به اتصال به شبکه ندارند.
۲. اپلیکیشن های مبتنی بر تعامل (Interactive transaction-based applications) این اپلیکیشن ها روی کامپیوتر های ریموت قرار دارند و کاربران از طریق کامپیوترهای خودشان به آن دسترسی دارند. شامل وب اپلیکیشن ها مانند فروشگاه های آنلاین می شود. یا می توان به وب اپلیکیشن های ایمیل نیز اشاره کرد. اپلیکیشن های تعاملی معمولا ترکیبی از داده های بزرگ را نگهداری می کنند که در هر تبادلی قابل دسترس است و به روزرسانی می شود.
۳. سیستم های کنترل تعبیه شده (Embedded control systems) : سیستم های کنترل نرم افزاری هستند که به کنترل و مدیریت دستگاه های سخت افزاری می پردازد.
۴. سیستم های پردازش دسته ای (Batch processing systems): سیستم‌های تجاری هستند که طراحی شده اند تا داده را در دسته های بزرگ پردازش کنند. تعداد زیادی از ورودی های منحصربفرد را پردازش کرده تا خروجی های مربوطه را ایجاد کنند. مانند سیستم های صدور صورت حساب دوره ای (مصلا شرکت مخابرات سیستمی برای صادر کردن قبض تلفن دارد)
۵. سیستم های سرگرمی (Entertainment systems) مانند بازی های کامپیوتری
۶. سیستم هایی برای مدل سازی و شبیه سازی (Systems for modeling and simulation) این سیستم ها معمولا توسط دانشمندان و مهندسین ساخته شده اند تا فرآیند های فیزیکی یا موقعیت هایی که شامل تعدادی اشیای جدا از هم و دارای فعل و انفعال هستند را شبیه سازی کنند
۷. سیستم های آنالیز و جمع آوری اطلاعات (Data collection and analysis systems) سیستم هایی هستند که از محیط خود داده جمع آوری می کنند و آن داده ها را برای پردازش، به سیستم دیگر می فرستند. این سیستم ها ممکن است که با سنسورها تعامل داشته باشند و اغلب درون محیطهای متخاصم مانند موتور ها نصب می شوند.
۸. سیستمی از سیستم ها، سیستم هایی هستند که در شرکت ها و سازمان‌های بزرگ دیگر استفاده می شوند که دارای تعدادی سیستم های نرم افزاری دیگر هستند.

البته مرز بین سیستمهای فوق واضح نمی باشد. مثلا اگر شما یک بازی برای موبایل بسازید.

هر یک از انواع سیستم ها تکنیک مهندسی نرم افزار مخصوص به خودش را می خواهد. برای مثال یک سیستم کنترلی تعبیه شده که برای خودرو استفاده می شود و ایمین برایش بحرانی است، باید روی ROM نوشته شود و درنتیجه تغییر دادنش بسیار هزینه بردار است. سیستم های این چنینی نیازمند به تایید و اعتبار سنجی گسترده ای دارند تا مراجعه خودرو پس از فروش برای رفع مشکل را به حداقل برسانند.

بررسی موردی

برای نشان دادن مفاهیم مهندسی نرم افزار، از چهار نوع سیستم به عنوان مثال، در ادامه استفاده شده است. این سیستم ها به شرح زیر هستند :

۱. سیستم تعبیه شده : تعریفش یه مقدار بالاتر آورده شده، فقط مسائل اینجور سیستم ها مربوط به سایز فیزیکی، واکنشگرا بودن و مدیریت مصرف برق و غیره می باشد. مثال برای سیستم تعبیه شده که در ادامه استفاده می کنیم، سیستم نرم افزاری است که پمپ انسولین را برای افرادی که دیابت دارند کنترل می کند.
۲. سیستم اطلاعاتی : هدف اصلی این نوع از سیستم ها مدیریت و فراهم کردن دسترسی به پایگاه داده ای از اطلاعات می باشد. مسایل مطرح در سیستم های اطلاعاتی : امنیت، قابل استفاده بودن، حریم خصوصی و حفظ یکپارچگی داده ها می باشد. مثال برای سیستمهای اطلاعاتی که در ادامه استفاده می کنیم، سیستم ذخیره اطلاعات پزشکی
۳. سیستم جمع آوری اطلاعات مبتنی بر سنسور : هدف اصلی اش جمع آوری اطلاعات از یک مجموعه سنسور و پردازش آنها به روشی می باشد. مسائل این نوع سیستم ها قابلیت اطمینان، امکان کارکردن در محیط های سخت و امکان نگهداری بالا می باشد. مثال ما از این سیستم ها : ایستگاه هواشناسی صحرایی
۴. محیط پشتیبان: مجموعه ای یکپارچه از ابزارهای نرم افزار می باشد که برای پشتیبانی تنوعی از فعالیت ها استفاده می شود. مثال ما «محیط یادگیری که یادگیری دانش‌آموزان در مدارس را پشتیبانی کند».

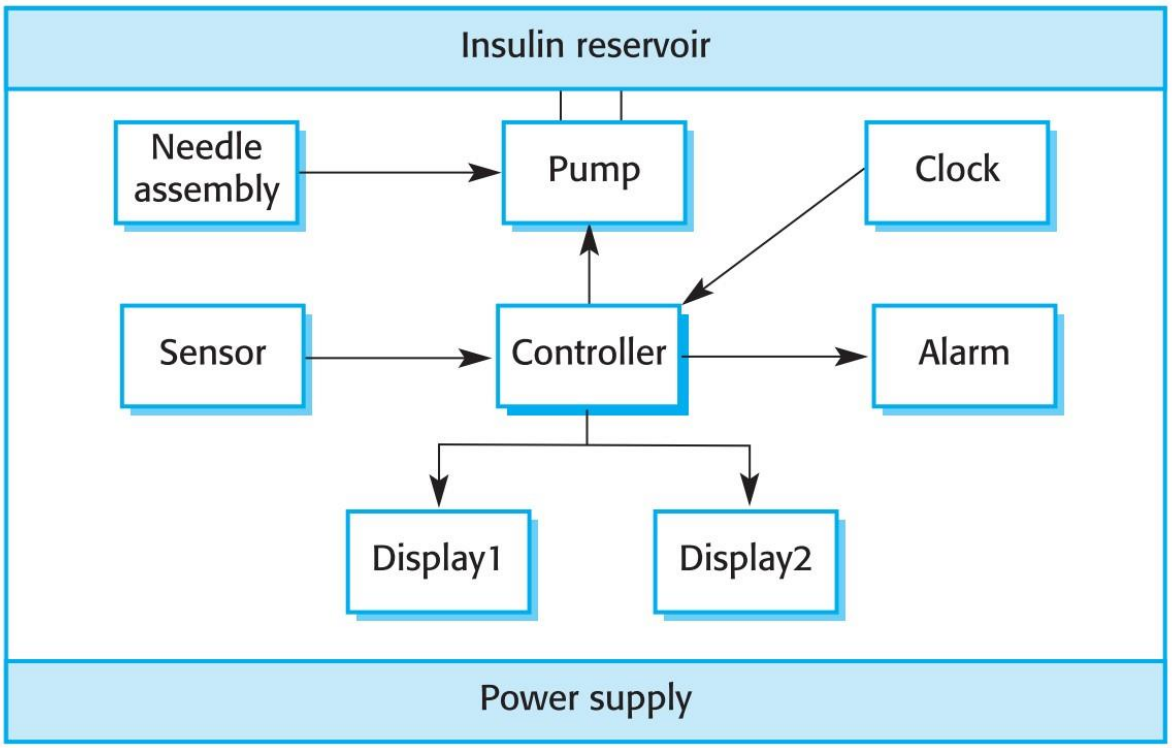
سیستم کنترل پمپ انسولین

در این سیستم تعبیه شده، اطلاعات از طریق سنسور محعآوری شده و پمپ برای تزریق میزان دوز مورد نظر کنترل می شود.

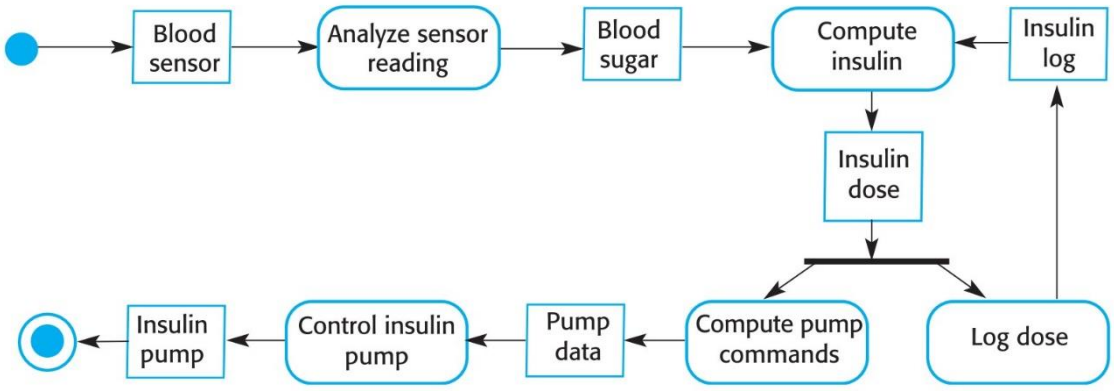
افراد مبتلا به دیابت از این سیستم استفاده می کنند. مسیه اصلی اینجا است که میزان تزریق انسولین نه تنها به مقدار قند خون بلکه به زمان آخرین تزریق نیز بستگی دارد. کاهش قند خون می تواند منجر به سکته مغذی و در بدترین حالت، منجر به مرگ شود. افزایش قند خون می تواند منجر به آسیب چشمی، آسیب کلیه و مشکلات قبلی شود.

سیستم تزریق انسولینِ کنترل شده توسط نرم افزار، از میکروسنسور تعبیه شده در بیمار جهت اندازه گیری پارامترهای خون استفاده می کند. این کنترلر مقدار خون را اندازه گیری می کند و مقدار انسولین مورد نیاز را نیز محاسبه می کند.. سپس اطلاعات رو به پمپ تزریق مینیاتوری جهت تزریق انسولین می فرستد.

تصویر ۱.۱ اجزای سخت افزاری و سازماندهی پمپ انسولین را نشان می دهد. نکته: به ازای هر پالس پمپ، یک واحد انسولین تزریق می شود، برای ۱۰ واحد انسولین باید ۱۰ پالس پمپ داشته باشیم. تصویر ۱.۲ نشان دهنده یک UML از Activity Model می باشد که نشان می دهد چگونه نرم افزار، سطح قند خون ورودی را به توالی از دستورات که منجر به حرکت پمپ انسولین می شوند، می فرستد.



تصویر ۱.۱



تصویر ۱.۲

به وضوح مشخص هست که این یک سیستم با بحران ایمنی می باشد. اگر پمپ نتواند عمل کند یا به درستی عمل نکند، در نتیجه اش سلامتی کاربر به خطر می افتد. این سامانه باید دو الزام سطح بالا را حتما داشته باشد:

۱. این سیستم برای تحویل انسولین در زمان لزوم، باید در دسترس باشد.
۲. این سیستم باید قابل اعتماد باشد و مقدار درستی از انسولین را برای متعادل نگه داشتن سطح قند خود تحویل دهد.

پس این سامانه باید طوری طراحی و پیاده سازی شود که اطمینان دهد دو الزام فوق همیشه رعایت می شود. بعدا در مورد این سامانه و چگونگی اطمینان از این الزامات و بحث بیشتر پیرامونش صحبت خواهیم کرد.

سیستم اطلاعات بیمار برای مراقبت از سلامت روان

یک سیستم اطلاعات پزشکی است که اطلاعات در مورد بیمارانی که از مشکلات سلامت روان رنج می برند و درمان هایی که دریافت کرده اند را حفظ می کند. اغلب بیماران سلامت روان نیاز به درمان حضوری در بیمارستان ندارند، اما، لازم است به طور مرتب در کلینیک های خصوصی شرکت کنند تا دکتر متخصص که از مشکلات آنها اطلاع دقیقی دارد آنها را معاینه کند.

سیستم مراقبت روان (شکل ۱.۳) یک سیستم اطلاعات بیمار است که برای استفاده در کلینیک ها در نظر گرفته شده است. از یک پایگاه داده مرکزی از اطلاعات بیماران ساخته شده اما طوری طراحی شده که روی لپتاپ اجرا شود، پس ممکن است در جایی که ارتباط شبکه امنی نداشته باشد، قصد دسترسی و استفاده از آن باشد. زمانی که شبکه محلی دارای دسترسی شبکه امن باشد، آنها از اطلاعات بیمار در پایگاه دادهها استفاده می کنند، و در زمانی که به شبکه متصل نیستند، اقدام به دانلود و استفاده از کپی های محلی اطلاعات بیمار می کنند. این سامانه یک سامانه کامل از اطلاعات همه بیماران برای همه بیماری ها نیست و صرفا تمرکز بر بیماران روح و روان دارد، البته می تواند با دیگر سیستم های اطلاعاتی کلینیک ارتباط داشته و تبادل اطلاعات کند.

این سیستم دو هدف دارد:

۱. تولید اطلاعات مدیریتی تا به مدیران خدمات سلامت اجازه دهد عملکرد را در برابر اهداف محلی و دولتی ارزیابی کنند.
۲. فراهم کردن اطلاعات به موقع برای کارکنان حوزه سلامت جهت پشتیبانی از درمان بیماران.

کاربران سیستم:

کارمندان مرکز پزشکی از جمله پزشکان، پرستاران و پرستاران در محل (ویزیت در منزل) می باشند. کاربران غیر کادر پزشکی : منشی (تنظیم قرار ملاقات و ...)، کارمند ثبت اسناد پزشکی و ادمین.

سیستم برای ذخیره اطلاعت بیماران (نام، آدرس، سن و ...)، مشاوره ها(تاریخ، پزشک معالج، برداشت های ذهنی بیمار و ...)، شرایط و درمان ها استفاده می شود. گزارشاتد در فواصل مشخص برای کارمندان پزشکی و مدیرات بهداشت و درمان تهیه می شوند. معمولا گزارشات مرتبط با کادر پزشکی تمرکز بر اطلاعات درباره بیماران مشخص می باشد، درحالیکه گزارش های مدیریتی ناشناس هستند و دغدغه هایشان مرتبط با شرایط، هزینه های درمان و ... می باشد.

ویژگی های کلیدی سیستم عبارت هستند از :

۱. **مدیریت مراقبت فردی** ، افراد کلینیک می توانند رکورد هایی برای بیماران بسازند، اطلاعات را در سیستم ویرایش کنند، تاریخچه بیماران را مشاهده کنند و ... . سیستم از خلاصه گیری اطلاعات پشتیبانی می کند، درنتیجه پزشکانی که از قبل بیمار را ندیده اند، می توانند به سرعت درباره مشکلات اساسی و درمان های که روی بیمار انجام شده، اطلاعاتی کسب کنند.
۲. **نظارت بر بیمار**، سیستم به طور مرتب بر پرونده بیمارانی که درگیر درمان هستند نظارت می کند و در صورت تشخیص مشکلات احتمالی، هشدارهایی صادر می کند. درنتیجه اگر یک بیمار برای مدتی پزشکی را ملاقات نکند، هشداری صادر خواهد شد. یکی از مهمترین المان های سیستم نظارت، ردیابی بیمارانی است که از بخش خارج شده اند و از انجام بررسی‌های قانونی در زمان مناسب اطمینان حاصل کند.
۳. **گزارشات مدیریتی**، سیستم به صورت ماهانه گزارش مدیریتی تولید می کند و تعداد بیماران درمان شده در هر کلینیک را نمایش می دهد، تعداد بیماران وارد شده که سیستم رو ترک کردند، تعداد بیماران مرخص شده، داروهای تجویز شده و هزینه ها.

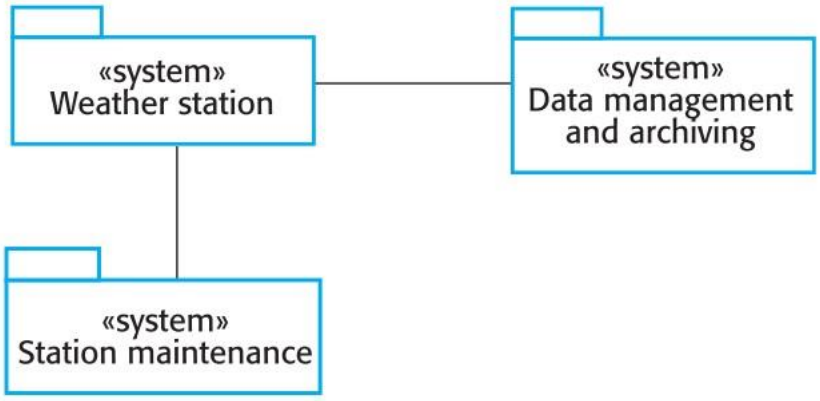
دو قانون مجزا روی سیستم تاثیر می گذارد: قانون حفاظت از داده که محرمانگی اطلاعات شخصی را کنترل می کند و قانون سلامت روان که بستری اجباری بیمارانی که برای خودشان یا دیگران خطر آفرین هستند را کنترل می کند. یکی از اهداف سیستم مراقبت روان این هست که اطمینان حاصل شود کارمندان همیشه مطابق قانون عمل می کنند و تصمیمات شان ذخیره می شود تا در صورت نیاز بعدا در مجامع قضایی بشود از آنها استفاده کرد.

در تمام سامانه های بهداشتی، حریم خصوصی از مهمترین و بحرانی ترین الزامات می باشد. این ضروری است که اطلاعات بیمار محرمانه باشد و بجز برای کادر پزشکی و خود بیمار برای هیچ فرد دیگری افشا نشود. این سامانه همچنین بحران-ایمنی را نیز دارد، بعضی از بیماری های روانی خطرناک هستند و منجر به خودکشی بیمار یا آسیب رساندن به بقیه می شود، در این مواقع سیستم باید به کادر پزشکی هشدار لازم را بدهد.

ایستگاه هواشناسی صحرایی

برای کمک به نظارت بر تغییرات آب و هوایی و بهبود دقت پیش بینی وضعیت آب و هوا در مناطق دورافتاده، دولت کشوری با مقادیر زیادی زمین های بیابانی تصمیم به پیاده سازی چند صد ایستگاه آب و هوایی در مناطق دورافتاده گرفته است. این ایستگاه ها داده ها را از مجموعه ای ابزارهای که دما و فشار، نور خورشید، سرعت باد و جهت باد را اندازه گیری می کنند، جمع آوری می کنند.

ایستگاه های آب و هوای صحرایی قسمتی از یک سیستم بزرگتر (تصویر ۱.۴) می باشند که سیستم اطلاعات آب و هوایی می باشد و داده های ایستگاه های هواشناسی رو جمع کرده و آنها را برای دسترسی دیگر سیستم ها به آنها و پردازش روی داده ها آماده می کند. سامانه ها در تصویر ۱.۴ عبارتند از:



تصویر ۱.۴

۱. سیستم ایستگاه هواشناسی، مسئولیت این سیستم جمع آوری داده های هواشناسی، اجرای برخی پردازش های اولیه داده و انتقال آن به سیستم مدیریت داده می باشد.
۲. سیستم مدیریت داده و آرشیو، این سیستم داده را از تمامی ایستگاه های هواشناسی جمع آوری می کند، پردازش داده و تحلیل را انجام داده و داده ها را در فرم ای که بتواند توسط دیگر سیستم ها، مانند سیستم های پیش بینی هوا استخراج شود، آرشیو می کند.
۳. سیستم نگهداری ایستگاه، این سیستم می تواند با استفاده از ماهواره با تمام ایستگاه های هواشناسی ارتباط برقرار کند و ناظر سلامتی این سیستم ها باشد و گزارشی از مشکلات را فراهم کند. می تواند نرم افزارهای تعبیه شده در این سیستم ها را بروزرسانی کند. در رویداد مواجهه با مشکل، این سیستم می تواند به صورت ریموت ایستگاه هواشناسی را کنترل کند.

در تصویر ۱.۴ از نمادهای پکیج UML جهت نشان دادن اینکه هر سیستم مجموعه ای از component ها می باشد استفاده شد و سیستم های جدا از هم با استفاده از کلیشه UMLبا نام «سیستم» مشخص می شود. ارتباط بین دسته ها نشان دهنده این است که تبادل اطلاعات وجود دارد اما، در این مرحله نیازی نیست تا آنها را با جزییات بیشتر تعریف کنیم.

همان طور که گفته شد، ایستگاه هواشناسی دارای ابزار های مختلفی برای اندازه گیری دما و فشار، نور خورشید، سرعت باد و جهت باد می باشد که در هر دوره ۲۴ ساعته انجام می شود. هر کدام از این ابزارها توسط یک سیستم نرم افزاری کنترل می شوند که به صورت دوره ای پارامتر ها را می خوانند و داده جمعآوری شده از این ابزارها را پیام می کند.

سیستم ایستگاه هواشناسی با جمعآوری مشاهدات هواشناسی در بازه های زمانی، عمل می کند. برای مثال دما هر دقیقه اندازه گیری می شود. اگرچه بدلیل اینکه پهنای باند به سمت ماهواره نسبتا کم می باشد. ایستگاه هواشناسی خودش مقداری پردازش محلی و تراکم داده ها را انجام می دهد. سپس زمانی که از سمت سیستم جمع آوری داده درخواستی آمد، این داده های تراکم شده را ارسال می کند. اگر هم امکان ارتباط میسر نبود، ایستگاه هواشناسی از داده ها به صورت محلی نگهداری می کند تا اینکه ارتباط برقرار شود.

هر ایستگاه هواشناسی توانش را از باطری می گیرد و باید بطور کامل جامع باشد، هیچ کابل شبکه یا شبکه برق خارجی وجود ندارد. کلیه ارتباطات از طریق یک اتصال ماهواره ای نسبتا کند انجام می شود، و ایستگاه هوایی باید مکانیسم هایی را برای شارژ کردن باطری خودش داشته باشد(انرژی خورشیدی یا بادی).

بدلیل استقرار در بیابان، آنها در معرض شرایط محیطی سخت قرار دارند و ممکن است حیوانات به انها آسیب بزنند. بنابراین نرم افزار ایستگاه، فقط به جمع آوری اطلاعات توجه ندارد، بلکه باید:

۱. روی ابزارها، برق، سخت افزارهای ارتباطی، خرابی گزارش به سیستم مدیریتی نظارت داشته باشید.
۲. برق سامانه را مدیریت کند، مطمئن شود که باطری ها در هر شرایط محیطی مجاز شارژ شوند و البته ژنراتورها در شرایط آب و هوایی بد، همچون بادهای شدید، خاموش شود.
۳. اجازه پیکربندی پویا را برای هنگامی که قسمتی از نرم افزار با نسخه جایگزین معاوضه می شود و زمانی که ابزارهای پشتیبان گیری در زمان خطای سیستمی به سیستم سوییچ می کنند را بدهد.

بدلیل آنکه ایستگاه هواشناسی باید دارای خود مختاری و بدون نظارت باشد، به این معنی است که نرم افزار نصب شده پیچیده است اگرچه جمع آوری داده عملا کاری ساده است.

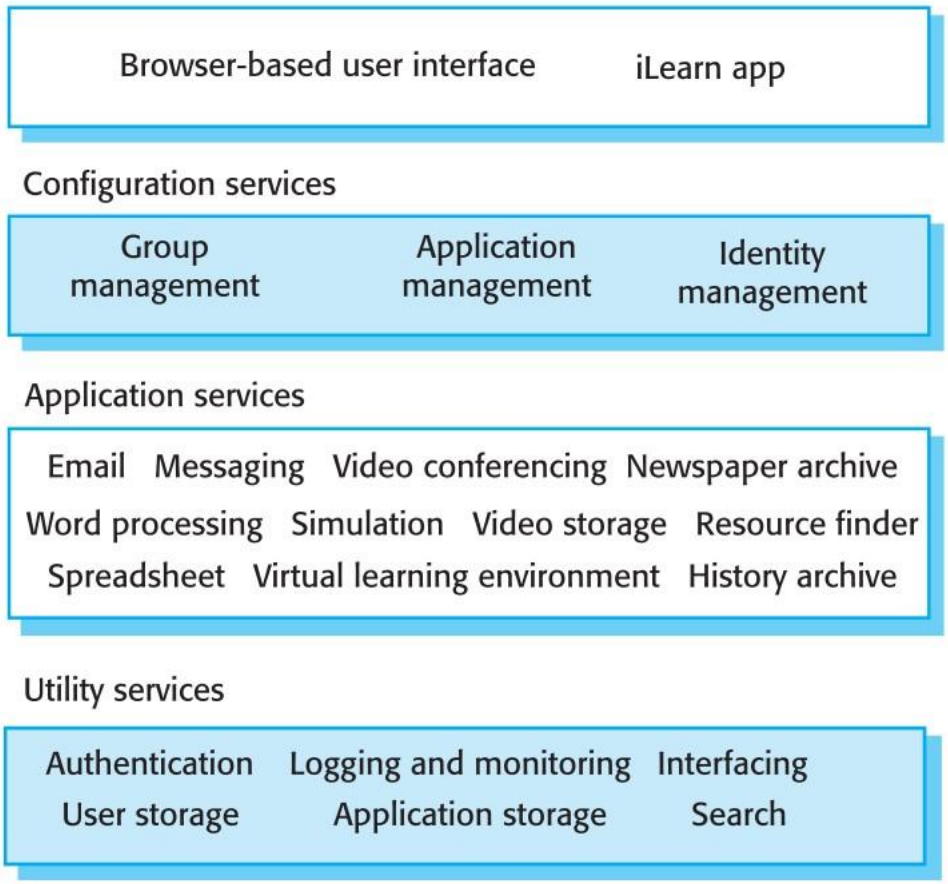
محیط یادگیری مجازی برای مدارس

خیلی از معلمان استدلال می کنند که استفاده از سامانه های نرم افزاری تعاملی جهت پشتیبانی از آموزش می تواند منجر به هر جفت بهبود انگیزش آموزنده ها و سطح عمیقی از دانش و درک دانش آموزان شود. گرچه توافق عمومی روی بهترین استراتژی برای یادگیری مبتنی بر کامپیوتر وجود ندارد و معلمین در عمل از گستره ی متفاوتی از ابزار های مبتنی بر وب جهت پشتیبانی یادگیری استفاده می کنند. ابزار استفاده شده مبتنی بر سن یادگیرنده ها، فرهنگ پیش زمینه شان، تجربه شان از استفاده کامپیوترها، تجهیزات در دسترس و ترجیحات معلم درگیر آموزش می باشد.

یک محیط یادگیری دیجیتال، چارچوبی است که ممکن است در آن مجموعه ای از اهداف-عام و ابزارهای طراحی خاص برای یادگیری تعبیه شده باشد، بعلاوه مجموعه ای از اپلیکیشن ها که برای نیاز دانش آموزان جهت استفاده از سیستم مجهز شده اند. چارچوب، سرویس های عامی همچون سرویس احراز هویت ، سرویس ارتباطات همگام و ناهمگام و سرویس ذخیره سازی را فراهم می کند.

ابزار های گنجانده شده در هر نسخه ای از محیط توسط معلمین و یادگیرنده ها انتخاب می شود تا نیازهای مختص آنها را بر طرف کند. اینجا می تونه اپلیکیشن ها عمومی همچون spread sheet یا اپلیکیشن های مدیریت یادگیری همچون محیط های یادگیری مجازی ((Virtual Learning Environment (VLE جهت مدیریت ارائه و ارزیابی تکالیف، بازی و شبیه سازی ها باشد. آنها همچنین می توانند شامل محتویات خاصی باشند، مثلا در باره جنگ داخلی آمریکا و اپلیکیشن هایی جهت دیدن و حاشیه نویسی محتویات.

تصویر ۱.۵ مدل معماری سطح بالا از محیط یادگیری دیجیتال ((iLearn)) می باشد که برای استفاده دانش آموزان بین ۳ تا ۱۸ سال در مدارس طراحی شده است. رویکرد اتخاذ شده این است که این یک سیستم توزیعی می باشد که تمام اجزای محیط سرویس هایی هستند که می توانند از هرجایی در بستر اینترنت در دسترس باشند. نیازی نیست که همه ی ابزاری های یادگیری با هم در یک مکان جمع شوند.



تصویر ۱.۵

سامانه مورد نظر، سامانه ای «مبتنی بر سرویس» می باشد که تمام اجزای سیستم یک سرویس قابل تعویض در نظر گرفته شده اند:

۱. سرویس ها کاربردی (Utility services) که عملکرد های مستقل از اپلیکیشن پایه را فراهم می کند و می تواند توسط دیگر سرویس های سیستم استفاده شود. سرویس های Utility معمولا به صورت مختص برای سیستم مورد نظر، توسعه و وفق داده می شوند.
۲. سرویس های اپلیکیشن (Application services) که اپلیکیشن های مشخصی مانند ایمیل، کنفرانس، اشتراک عکس و ... را فراهم می کند. و به محتویات آموزشی مشخصی مانند فیام های علمی و منابع تاریخی دسترسی دارد. سرویس های اپلیکیشن، سرویس های خارجی هستند که بطور اختصاصی برای سیستم خریداری شده اند یا به صورت رایگان در اینترنت در دسترس هستند.

۳. خدمات پیکربندی، استفاده می شوند تا محیط را با مجموعه مشخصی از سرویس های اپلیکیشن وفق دهند و تعریف کنند سرویس ها چگونه بین دانش آموزان، معلمین و پدرومادر ها به اشتراک گذاشته شوند.

محیط به گونه ای طراحی شده که سرویس ها بتوانند با سرویس های جدید که در دسترس هستند جایگزین شوند و نسخه های متفاوت از سیستم را که با سن کاربر متناسب هستند را فراهم کند. به این معنی است که سیستم باید از دو سطح از ادغام سرویس ها پشتیبانی کند:

۱. سرویس های یکپارچه (Integrated services)، سرویس هایی هستند که یک API (application programming interface) را پیشنهاد می دهند و می توانند توسط دیگر سرویس ها از طریق API در دسترس باشند. ارتباط مستقیم سرویس-به-سرویس در نتیجه ممکن می شود. سرویس ارتباطی مثالی از سرویس یکپارچه می باشد. به جای استفاده از مکانیسم های احراز هویت شان، یک سرویس احراز هویت میتواند توسط سرویس های دیگر برای احراز هویت کاربران فراخوانی شود. اگر کاربر احراز هویت شده است، سپس سوری احراز هویت ممکن اسن که اطلاعات احراز هویت را مستفیما به سرویس دیگری انتقال دهد، توسط یک API، بدون اینکه نیاز باشه کاربر مجدد احراز هویت را انجام دهد.
۲. سرویس های مستقل (Independent services)، سرویس هایی هستند که به سادگی از طریق واسط مرورگر در دسترس هستند و مستقل از دیگر سرویس ها عمل می کنند. اطلاعات فقط توسط عملهای صریح کاربر (مانند copy یا paste) با دیگر سرویس ها به اشتراک گذاشته می شود، احراز هویت مجدد اختمالا برای هر سرویس مستقلی مورد نیاز باشد.

اگر یک سرویس مستقل به طور گسترده استفاده شود، تیم توسعه ممکن است آن سرویس را یکپارچه کنند در نتیجه یک سرویس پشتیبانی شده و یکپارچه می شود.

نکات کلیدی

- مهندسی نرم افزار یک سری اصول مهندسی می باشد که دغدغه اش تمامی جنبه های تولید نرم افزار می باشد.
- نرم افزار تنها یک برنامه یا چند برنامه نمی باشد، اما همچنین شامل تمامی اسناد الکترونیکی که مورد نیاز کاربران سیستم، کارمندان اطمینان از کیفیت و توسعه دهندگان هست، می باشد. ویژگی های اساسی محصول نرم افزاری قابلیت نگهداری، قابلیت اطمینان و کارایی امنیت و قابل قبول بودن می باشد.

- فرآیندهای نرم افزار شامل تمامی فعالیت های مربوط به توسعه نرم افزار می باشد. فعالیت های سطح بالا از مشخصات، توسعه، اعتبارسنجی و تکامل بخشی از کلیه فرآیند های نرم افزاری است.
- انواع مختلفی از سیستم وجود دارد که هر کدام نیاز به مهندسی نرم افزار، ابزار و تمنیک های توسعه مناسب خود دارد.
- ایده های اساسی مهندسی نرم افزار قابل استفاده روی تمامی انواع سیستم های نرم افزاری است. این مبانی شامل فرآیند های نرم افزار مدیریت شده، قابلیت اعتماد و امنیت نرم افزار، مهندسی الزامات و استفاده مجدد از نرم افزار می باشد.
- مهندسین نرم افزار مسئولیت جامعه و حرفه مهندسی را دارند. آنها نباید به سادگی نگران مسائل تکنیکال باشند، اما باید در مورد مسائل اخلاقی که روی کارشان تاثیر می گذارند، آگاهی داشته باشند.

تا اینجای کار من سعی کردم خلاصه ای از فصل اول کتاب «مهندسی نرم افزار» نوشته سامرویل ، ویرایش ۱۰م را با شما به اشتراک گذاشته باشم. در اینجا می توانید اسلاید مربوط تا اینجا را ببینید.