بسمه تعالی

مهندسی نرم افزار

استاد:

مهندس عالیه همتی

آدرس وب سایت:

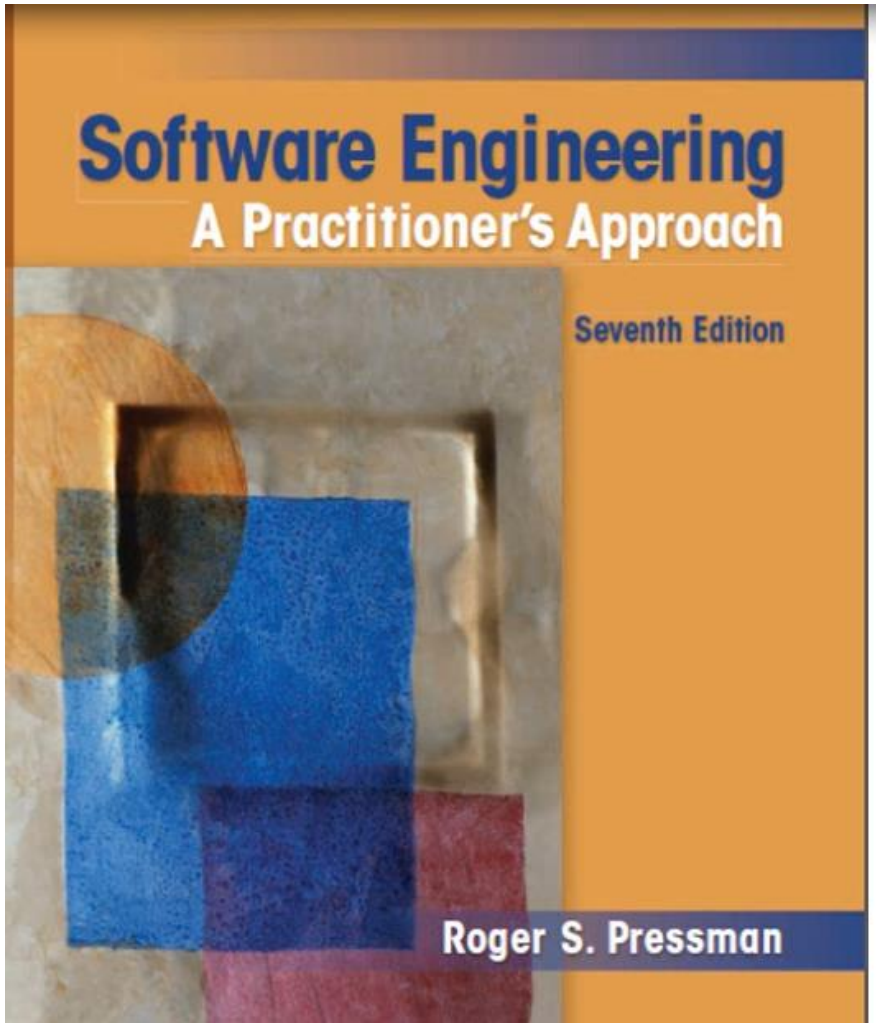
www.ahemati.ir

آدرس پست الکترونیکی:

Aalia.hemmati92@gmail.com

بهمن ۱۴۰۰

منبع



راجر اس. پرسمن

منبع ۱: کتاب مهندسی نرم افزار

مولف: راجراس.پرسمن

مترجم ۱: دکتر محمد مهدی سالخورده حقیقی

یا مترجم۲ : نوید هاشمی طبا

بارم بندی درس مهندسی نرم افزار

- ✓ ۱۳ نمره پایان ترم
- ✓ ۲ نمره – کار تحقیقاتی-ارائه تحت پاورپوینت
- ✓ ۵ نمره پروژه- کارعملی شامل:

بخش ۱: ارتباطات-برنامه ریزی (تحت پاورپوینت/ ۱نمره)+

بخش ۲: طراحی نرم افزار (توسط رشنال رز/ ۴ نمره)

موضوعات کار تحقیقاتی-ارائه

۱- فرآیند RUP , EUP

۲- متدلوژی SSADM

۳- تفاوت ماژول و مولفه و وب سرویس

۴- XP(Extreme programming)

۵- MS Project

۶- برنامه نویسی یکپارچه

۷- تابع گرا (ساخت یافته)

۹- مولفه گرا

۱۰- سرویس گرا

۱۱-SSPI

موضوعات کار تحقیقاتی-ارائه

۱۲-مدیریت پیکربندی نرم افزار

۱۳- طراحی معماری نرم افزار

۱۴- طراحی رابط کاربر

۱۵- تکنیک های آزمایش نرم افزار

۱۶- معیارهای تکنیکی برای نرم افزار

۱۷- مفاهیم و اصول طراحی

۱۸- مدلسازی تحلیل

موضوعات پیشنهادی برای پروژه

۱- سیستم تعمیرگاه(بخش پذیرش- حسابداری- تعمیرات – مدیریت.....)
۲- بیمارستان (مثلا بخش نوبت دهی و پذیرش و .....)
۳- انبارداری(ورود کالا به انبار- خروج کالا ار انبار- انتقال ار انبار به انبار – کنترل موجودی...)
۴- حسابداری(گردش مالی- بدهکاری و بستانکاری و ..)
۵- فروشگاه(بخش مدیریت - فروش (فاکتور فروش – فاکتور مرجوعی از فروش) - خرید(فاکتور خرید- فاکتور مرجوعی از خرید)
۶- سیستم دانشگاه(بخش ثبت نام و انتخاب واحد- امور مالی- آموزش و ...)

دانلود و نصب ابزارهای مورد نیاز

۱) نرم افزار مدلسازی رشنال رز

IBM Rational Rose Enterprise

از طریق لینک زیر دانلود کنید:

https://soft98.ir/

۲) نرم افزار Visual Studio 2012 از طریق لینک زیر دانلود کنید:

https://visualstudio.microsoft.com/downloads/

https://soft98.ir/software/programming.html

بخش اول:
مهندسی نرم افزار
و مدل های فرآیند تولید و توسعه نرم افزار

فصل اول
نرم افزار و مهندسی نرم افزار

فهرست مطالب

۱-نرم افزار کامپیوتر

- محصولی است که مهندسین نرم افزار آن را طراحی و تولید می کنند ، شامل :
  - دستوراتی است که در یک کامپیوتر با هر اندازه و ساختاری اجرا می شوند.
  - داده هایی است شامل ترکیبی از اعداد و متن و اطلاعات تصویری ، ویدئویی و صوتی
  - مستنداتی است حاوی متن ها و فرم ها
- دلیل اهمیت آن تاثیری است که در تجارت، فرهنگ، صنعت و ... می گذارد.
- برای ساخت چنین محصولی با چنین تاثیری نیاز به یک شیوه مهندسی است که به آن مهندسی نرم افزار می گویند.

۲- خصوصیات نرم فزار و سخت افزار

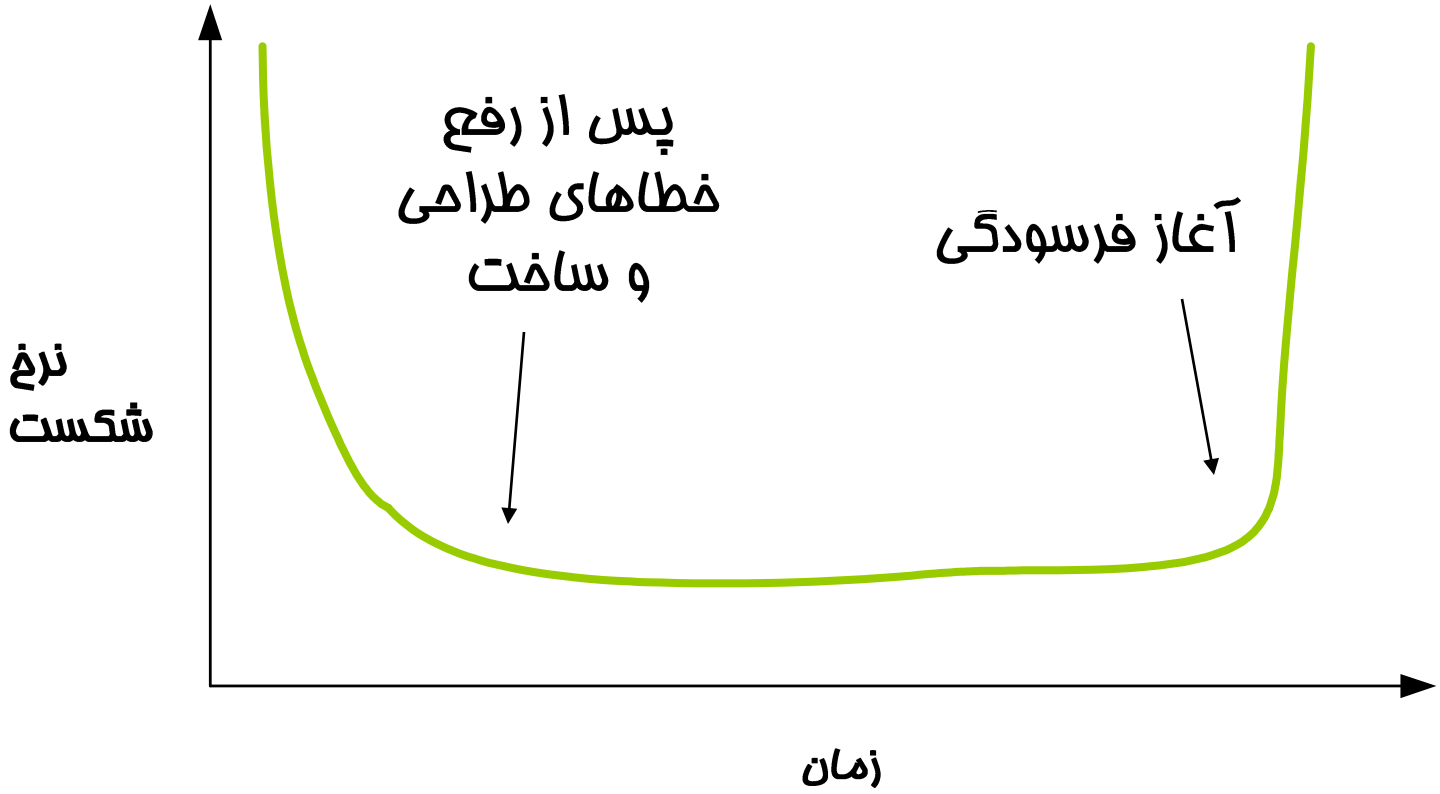
نرم افزار بیشتر یک عنصر منطقی است تا یک سیستم فیزیکی بنابراین دارای ویژگی هایی است که تا حد زیادی از مشخصه های یک سخت افزار متفاوت است .

۱.نرم افزار توسعه می یابد یا طراحی می شود، اما به مفهوم کلاسیک ساخته نمی شود.

- ✓ در هر دو آنها کیفیت بالا حاصل طراحی خوب خواهد بود، اما مرحله ساخت در مورد سخت افزار می تواند یک سری مشکلات کیفی داشته باشد که در مورد نرم افزار وجود ندارد یا به راحتی قابل حل باشند.
- ✓ هر دو مستلزم ساختن یک محصول هستند اما روش ها متفاوتند.

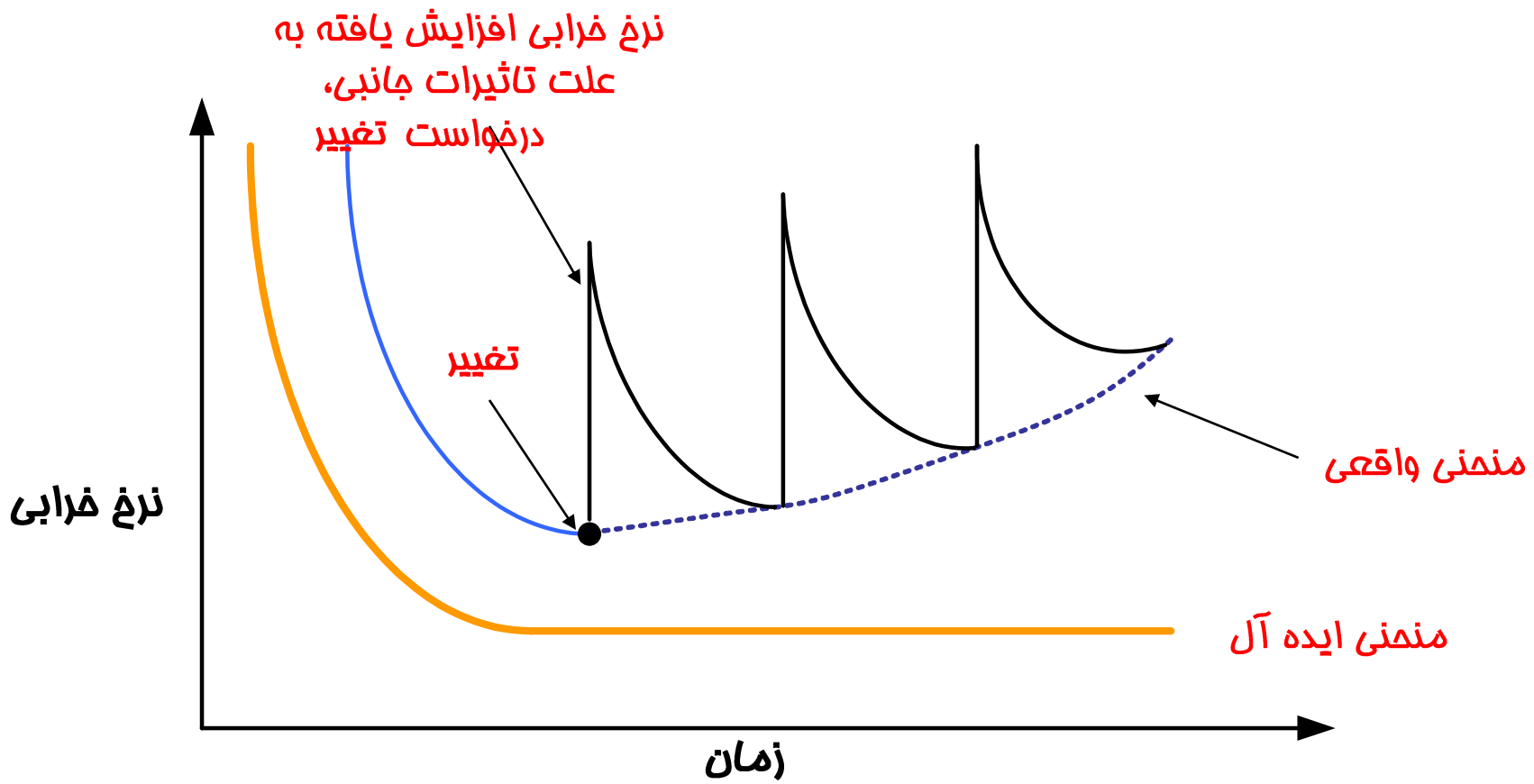
۲.سخت افزار بعد از مدتی کارایی خود را از دست می دهد و فرسوده می شود ولی نرم افزار فرسوده نمی شود اما کیفیت خود را از دست می دهد.

سخت افزار در اوایل عمرش میزان عدم موفقیت نسبتاً بالایی دارد. نقایص اصلاح شده و میزان شکست برای مدتی به سطح ثابتی میرسد و با گذشت زمان، سخت افزار فرسوده می شود و نرخ شکست دوباره افزایش می یابد.



منحنی نرخ شکست سخت افزار نسبت به زمان

نرم افزار در معرض عوامل محیطی (دما – رطوبت ...)که سخت افزار را خراب می کند نمی باشد . بنابراین از نظر تئوری منحنی شکست نرم افزار، به شکل یک منحنی ایده آل می باشد.



منحنی نرخ خرابی واقعی نرم افزار نسبت به زمان

۳. گرچه صنعت به سمت مونتاژ اجزا پیش می رود اما نرم افزارهمچنان سفارشی تولید می شوند.

نکته: در جهان سخت افزار ، استفاده مجدد از قطعات ، بخش متداول فرآیند مهندسی است.

در جهان نرم افزار این روش به طور گسترده تر استفاده می شود. بنابراین مهندس می تواند روی بخش های جدید تمرکز کند.(مثال ایجاد کتابخانه ها، رابط های گرافیکی کاربر با استفاده از اجزای قابل استفاده مجدد)

۳- دامنه های کاربرد نرم افزار

۱) سیستمی : system software

۲) کاربردی : application software

۳) مهندسی/علمی : engineering/scientific software

۴) تعبیه شده : embedded software

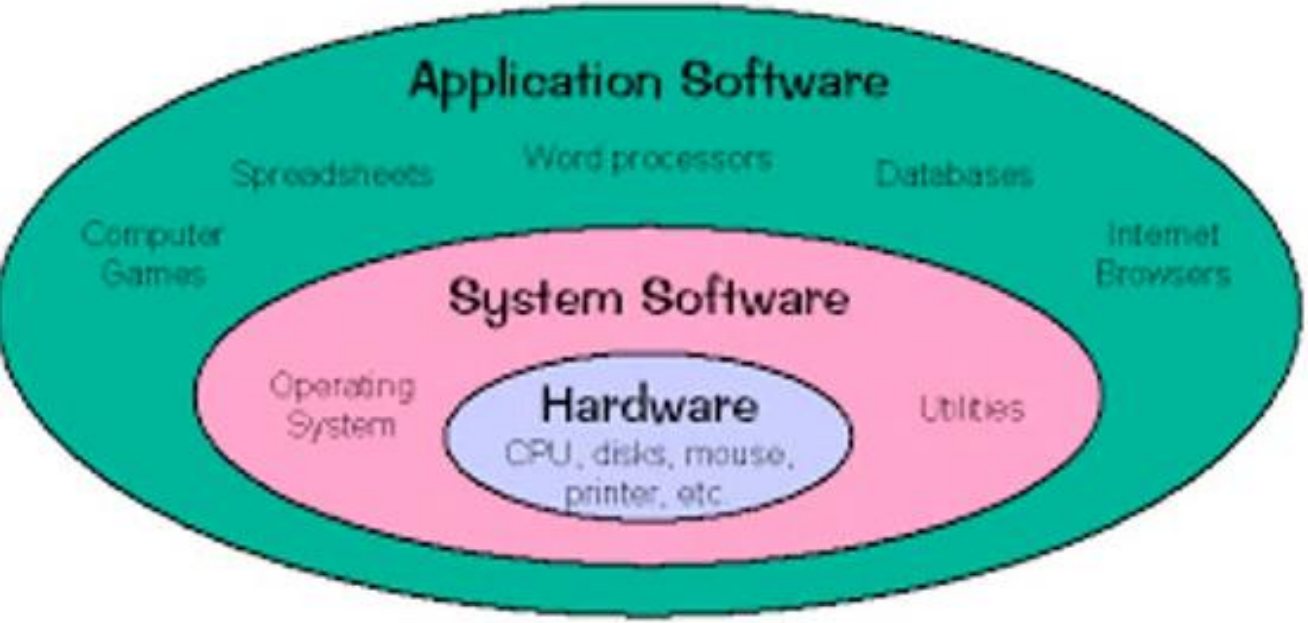
۵) خط تولید : product-line software

۶) برنامه های کاربردی تحت وب: WebApps

۷) هوش مصنوعی: AI software

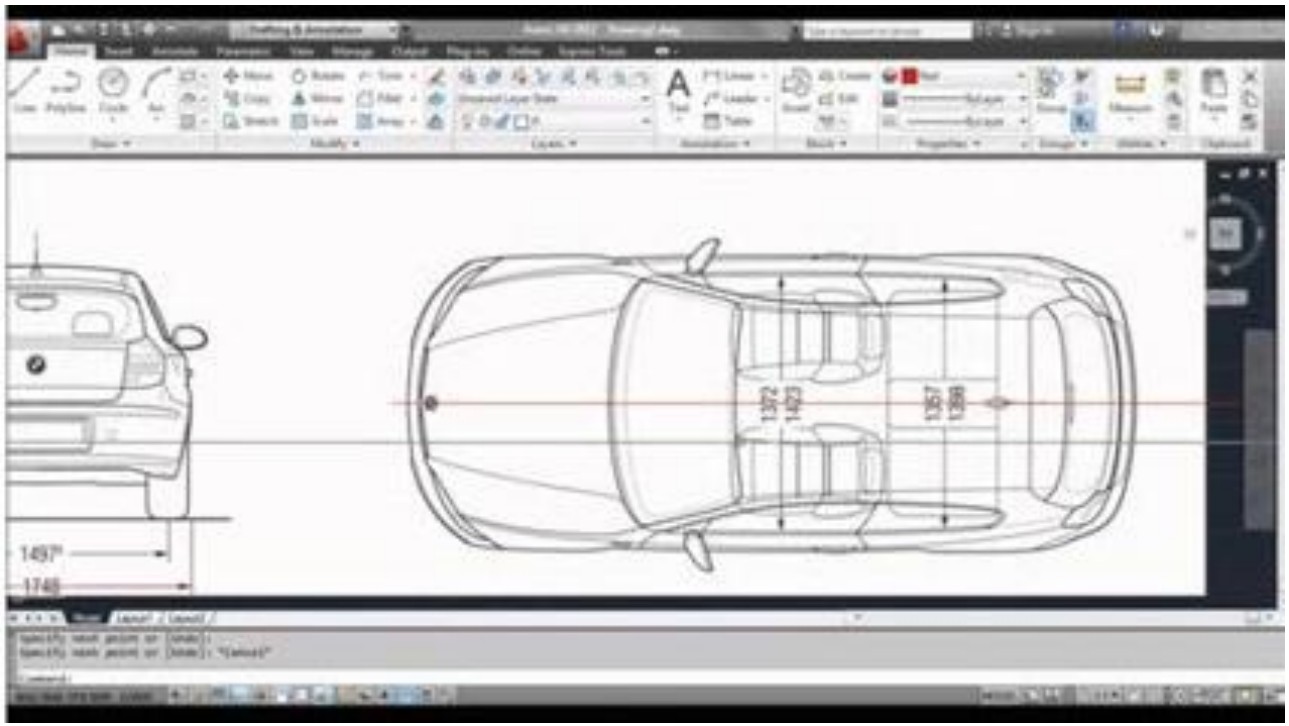
دامنه کاربردهای نرم افزار

۱- نرم افزارهای سیستمی: مجموعه ای از برنامه هاست که برای سرویس دهی به برنامه های دیگر نوشته شده اند. مثل : کامپایلرها(تبدیل زبان سطح بالا به زبان سطح پایین یا زبان ماشین) ویراستارها وسیستم عامل (android-windows,..)و...



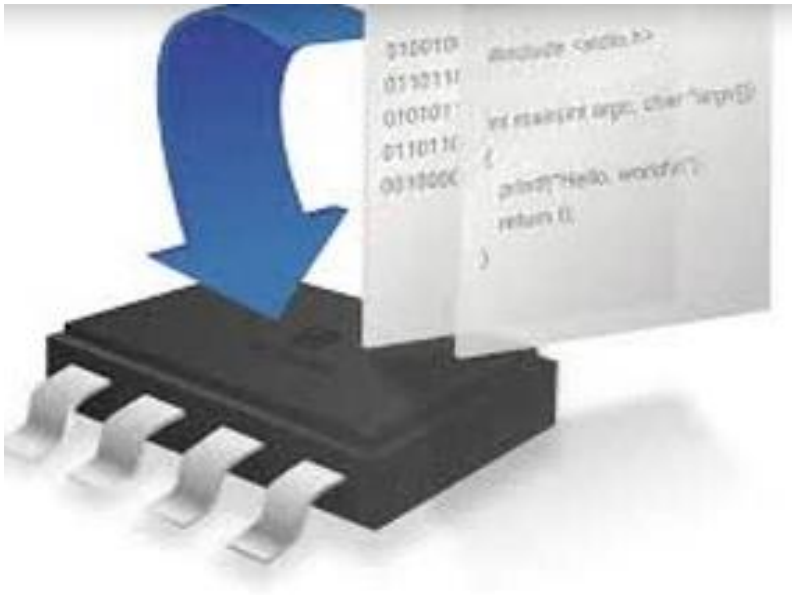
۲) نرم افزارهای کاربردی:
برنامه های مستقل که یک نیاز تجاری را برطرف می سازد.

۳-نرم افزارهای مهندسی و علمی : توسط الگوریتم هایی مشخص می شوند که "ارقام و اعداد" را پردازش می کنند. کاربردهایش در شبیه سازی سیستم ها، ستاره شناسی، شناخت آتشفشان، تحلیل فشار ،خط تولید خودکار و ...



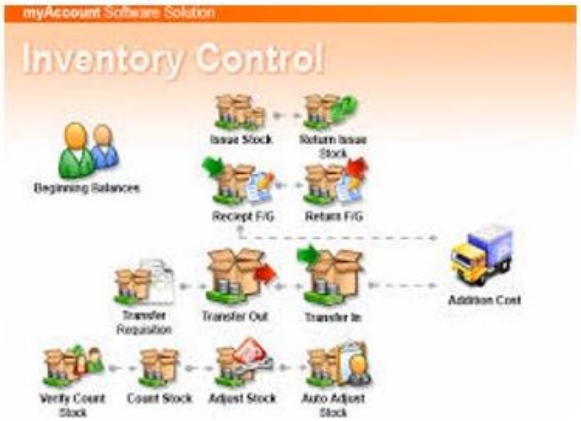
٤-نرم افزارهای تعبیه شده : نرم افزار قرار داده شده در ابزار صنعتی که ابزار را کنترل می کند و درحافظه فقط خواندنی جای دارد. (محصولات هوشمند)

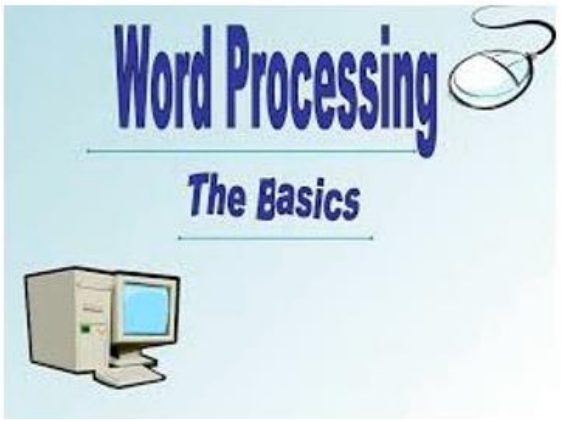
- برای مثال: توابع دیجیتالی در اتومبیل مانند کنترل سوخت، صفحه نمایش داشبورد، سیستم های ترمز....



۵) نرم افزارهای خط تولید : product-line software

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





- ✓ بازار مشتریان خاص و محدود

٦-نرم افزار های تحت وب:

صفحات وب که توسط مرورگرها بازیابی می شوند نرم افزارهایی هستند شامل دستورات اجرایی همچون HTML, JAVA,PHP و داده ها مثل قالب های متعدد تصویری و صوتی.

۷) نرم افزار های هوش مصنوعی: AI software

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

مثال هایی از کاربرد :

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

۴- بحران نرم افزار

بحران نرم افزار برای اولین بار در کنفرانس مهندسی نرم افزار ناتو (۱۹۶۸) مطرح شد که تولید و توسعه سیستمهای نرم افزاری دشوار است.

عوامل اصلی این بحران عبارتند از :

۱. هزینه بالای تولید نرم افزار
۲. تاخیر در تولید و تحویل نرم افزار
۳. کیفیت پایین نرم افزار
۴. نگهداری پرهزینه نرم افزار
۵. پیشرفت سریع سخت افزار
۶. افزایش پیچیدگی محصولات

بحران نرم افزار

یک راه حل مؤثر، استفاده از روش توسعه مبتنی بر مؤلفه جهت تولید مؤلفه و توسعه سیستمهای مبتنی بر مؤلفه است با مونتاژ مؤلفه های پیش ساخته با قابلیت استفاده مجدد می باشد.

برای مقابله با بحران، مهندسی نرم افزار مطرح شده است.

نسلهای پنجگانه تولید و توسعه سیستمهای نرم افزاری عبارتند از :

۱- برنامه نویسی یکپارچه

۲- تابع گرا (ساخت یافته)

۳- شئی گرا

۴- مولفه گرا

۵- سرویس گرا

مهندسی نرم افزار با گذر از نسلهای پنجگانه تولید و توسعه سیستمهای نرم افزاری و با کسب دانش و تجربه از فنآوریهای نوین و ابتکاری، پیشرفتهای چشمگیری داشته است.

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.

پروژه موفق

پروژه موفق یعنی پروژه در زمان پیش بینی شده یا مناسب به پایان برسد.

۱. با بودجه تخمین زده شده(مورد موافقت).
۲. زمان پیش بینی شده (زمان مناسب).
۳. کیفیت خوب و مناسب طرح.

۵- مهندسی نرم افزار

- تعریف مهندسی نرم افزار بنا بر پیشنهاد انجمن IEEE :

مهندسی نرم افزار عبارت است از بکارگیری یک روش سیستماتیک، منظم و قابل اندازه گیری برای تولید و توسعه ، عملیاتی کردن و نگهداری نرم افزار .
به عبارت دیگر بکارگیری اصول مهندسی در تولید نرم افزار.

- تعریف مهندسی نرم افزار توسط Fritz Bauer :

ایجاد و استفاده از اصول ساده مهندسی به منظور رسیدن به یک نرم افزار مقرون به صرفه که قابل اطمینان بوده و بر روی دستگاههای واقعی کارآمد باشد.

۶-تفاوت مهندسی نرم افزار و علوم کامپیوتر

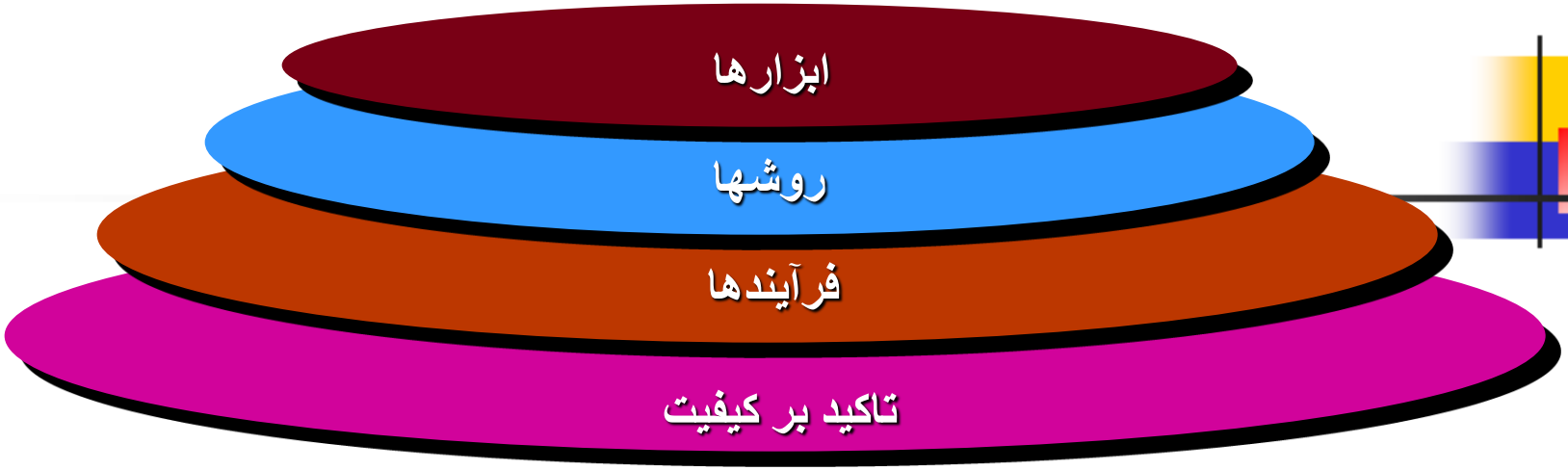
علوم کامپیوتر به تئوری ها و قضایایی که مربوط به نرم افزار می باشد، می پردازد ولی در حالی که مهندسی نرم افزار به جنبه های عملیاتی کردن آن تئوری ها می پردازد.

۷-تفاوت مهندسی نرم افزار و مهندسی سیستم

مهندسی سیستم به بررسی کلیه ی جنبه های یک سیستم هم از نظر نرم افزاری و هم سخت افزاری می پردازد ولی مهندسی نرم افزار به فکر عملیاتی کردن فرآیندهای نرم افزاری است.

فصل دوم
فرایند و مدل های فرآیند

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.

مانند ابزار Rational Rose که در تکنیک‌های تحلیل و طراحی بسیار مفید است و یا NUnit مربوط به تکنیک‌های تست نرم‌افزار می‌شود که توسط برنامه‌نویس انجام می‌شود و صدها برنامه کاربردی دیگر.

تعریف فرایند

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

به مجموعه ای از فعالیت ها ونتایج مربوط به آنها که منجربه تولید محصول نرم افزاری می گردد فرایند نرم افزاری گفته می شود.

چارچوب فرایند کلی برای تولید محصول نرم افزاری

چارچوب فرایند کلی برای مهندسی نرم افزار شامل فعالیت های زیر می باشد شامل:

۱-ارتباطات communication(شناسایی نیازمندیها)

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

دراین فاز مشخص میشود که سیستم چه خدماتی را باید ارائه دهد در واقع نیازهای مشتری شناسایی می شود .

از طرق زیر عملیات جمع آوری اطلاعات و شناخت نیازمندی ها صورت می گیرد:

- ✓ مشاهده
- ✓ پرسشنامه
- ✓ مصاحبه با کاربران
- ✓ بررسی اسناد و مدارک
- ✓ بررسی سیستم های موجود
- ✓ بررسی سایر سیستم های مشابه
- ✓ صورت برداری منابع مورد نیاز
- ✓ تجزیه و تحلیل خروجی مورد نیاز
- ✓ نمونه گیری و تجزیه و تحلیل آماری

۲- برنامه ریزی(planning)

در این مرحله عملیاتی مثل توصیف وظایف فنی و ترتیب فعالیت ها که قرار است اجرا شوند، مدیریت ریسک، منابعی که مورد نیاز خواهند بود(انتخاب نفرات تیم، زمان، امکانات سخت افزاری و نرم افزاری، بودجه)، زمانبندی کاری و نوشتن گزارش ها و محصولاتی که باید تولید شوند برنامه ریزی می شوند.

۳- طراحی(Modeling)

مهندس نرم افزار جهت درک بهتر نیازها مدلسازی و طراحی می کند مثل رسم نمودارها ی use case به زبان UML با استفاده از ابزار رشنال رز

۴-ساخت Construction(پیاده سازی و تست):

در این مرحله، کدنویسی و تست های لازم برای یافتن خطاهای موجود در کدها انجام می شود.

مراحل تست:

- ✓ تست واحد: پس از نوشتن یک تابع یا Procedure توسط برنامه نویس با ورودی های مختلف تست می شود.
- ✓ تست ماژول: یک ماژول تشکیل شده از تعدادی تابع که باهم تعامل دارند، پس از نوشتن یک ماژول ، برنامه نویس موظف است این واحد ها را به صورت ترکیبی چک کند.
- ✓ تست زیر سیستم : پس از آنکه چند ماژول باهم ترکیب شده و تشکیل زیر سیستم را می دهد بایستی رفتار زیر سیستم بررسی شود.
- ✓ تست سیستم: از چندین زیر سیستم تشکیل شده که باهم تعامل دارند و باید چک و بررسی شود.
- ✓ تست آلفا: در این قسمت سیستم های تولید شده توسط داده های مشتری تست می گردد وآنقدر ادامه می یابد تا مشتری سیستم را قبول کند.
- ✓ تست بتا: بررسی یا تست یک سیستم در اثر واگذاری به مشتری های مختلف و کار کردن آنها با سیستم توسط داده های واقعی.

۵-استقرار Deployment(تکامل و پشتیبانی)

نرم افزار به مشتری تحویل داده می شود تا محصول را ارزیابی و براساس آن مهندس نرم افزار باید مشکلات نرم افزار را برطرف نماید و خطاهای موجود را اصلاح کند و درصورت نیاز پس از گذشت مدتی اگر کاربر درخواست توسعه ی نرم افزار را داشت و یا توقع دیگری از نرم افزار داشت باید تغییرات لازم جهت تولید نرم افزار جدید اعمال گردد.

مرحله ی پشتیبانی روی تغییر متمرکز است.

در طول این مرحله چهار نوع تغییر را شاهد هستیم:

- ✓ الف) اصلاح: نواقص موجود را برطرف کند.
- ✓ب) تطابق: تطبیق نرم افزار با محیط خود.
- ✓ج)بهبود وضعیت: اضافه کردن فعالیت جدیدی در سیستم
- ✓د) پیشگیری: پس از گذشت زمان ، نرم افزار سودمندی خود را از دست می دهد و نیاز به مهندسی مجدد نرم افزار می باشد.

فعالیت فوق با یکسری فعالیتهای چتری (Umbrella activities) که در اسلاید بعدی نامبرده تکمیل می گردد.

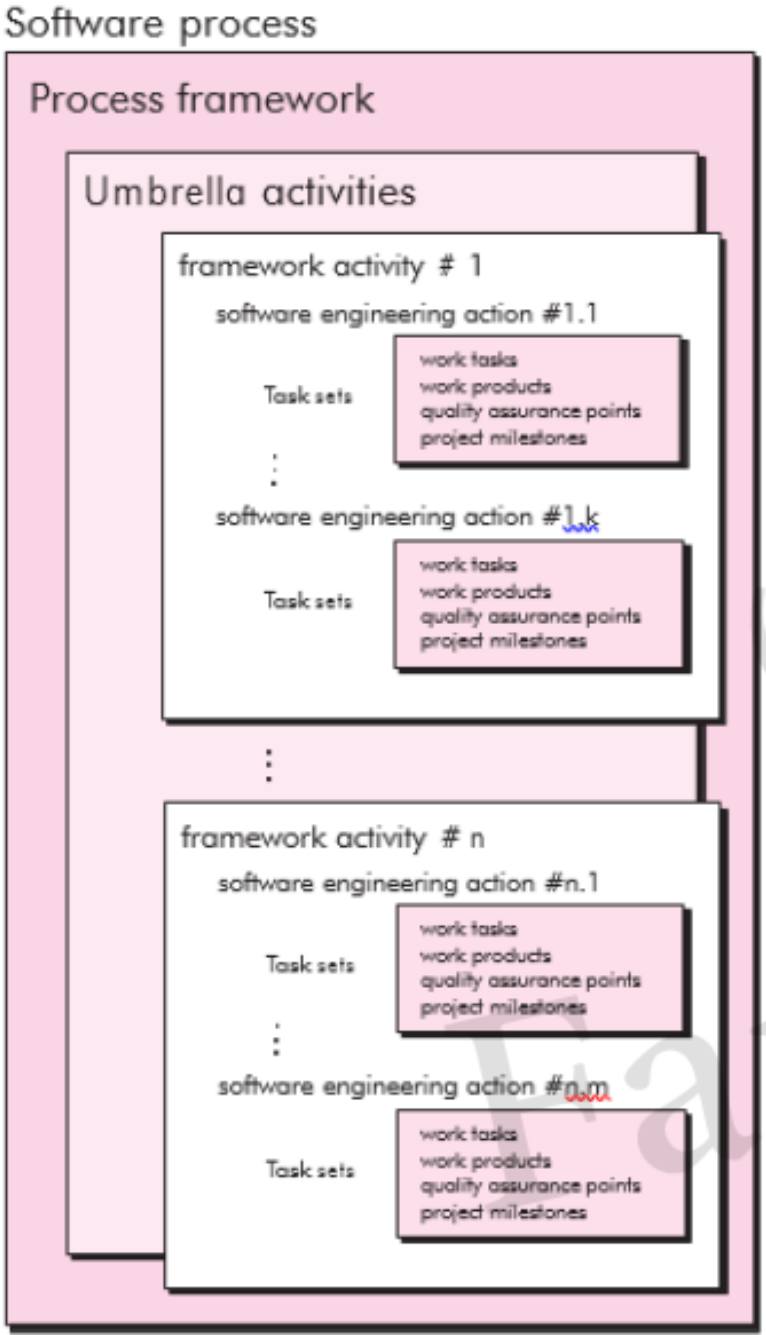
فعالیت های چتری:

فعالیت هایی هستند که در طول پروژه بصورت مداوم تکرار می شوند و هنگامی به پایان می رسند که کل پروژه به پایان رسیده باشد.

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)

یک مدل فرایند کلی



work tasks
work products
quality assurance points
project milestones

به چه وظایفی باید عمل شود
چه محصولاتی باید تولید شود
به چه نقاط تضمین کیفیتی نیاز می باشد
چه نقاط عطفی برای نشان دادن پیشرفت فرایند به کار گرفته می شود.
نقاط عطف ، نقاط پایانی یک فعالیت می باشد که معمولا یک خروجی به صورت رسمی یا غیر رسمی داریم.

جریان فرآیند

جریان فرایند نشان می دهد که فعالیت های چتری و وظایف و کنش هایی که در داخل فعالیت چارچوبی رخ می دهند از نظر زمانی چگونه سازماندهی می شوند.

انواع جریان فرایند:

۱- خطی

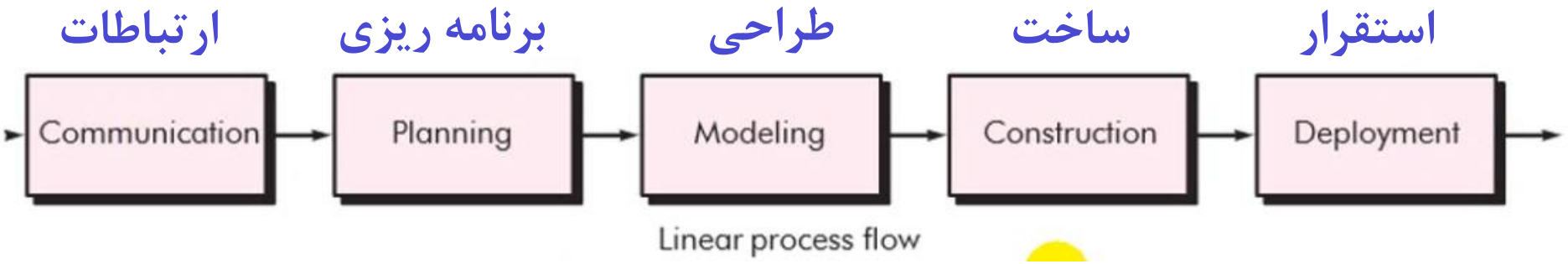
۲- مبتنی بر تکرار

۳- تکاملی

۴- موازی

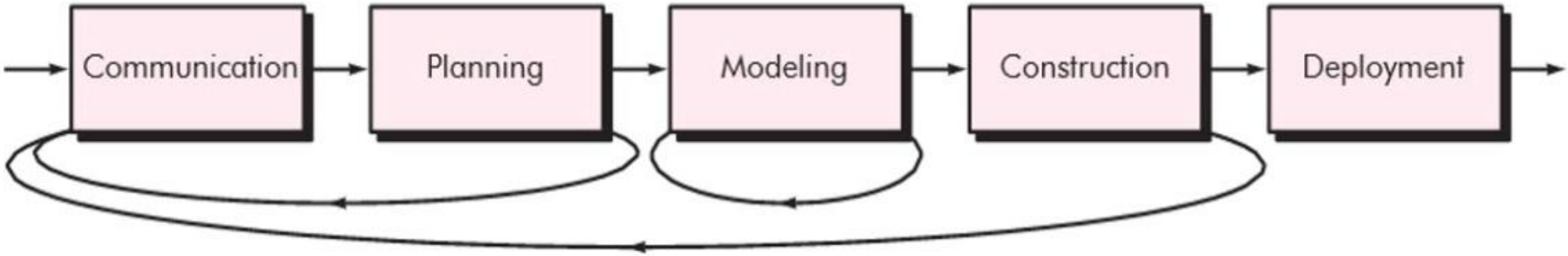
۱- جریان فرآیند خطی

هرکدام از ۵ فعالیت فرایند کلی نرم افزار به ترتیب اجرا می شود، به طوری که با ارتباطات آغاز و به استقرار ختم می شود.



۲- جریان فرآیند مبتنی بر تکرار

در این جریان پیش از رفتن به دور تکرار بعدی یک یا چند فعالیت تکرار میشود.

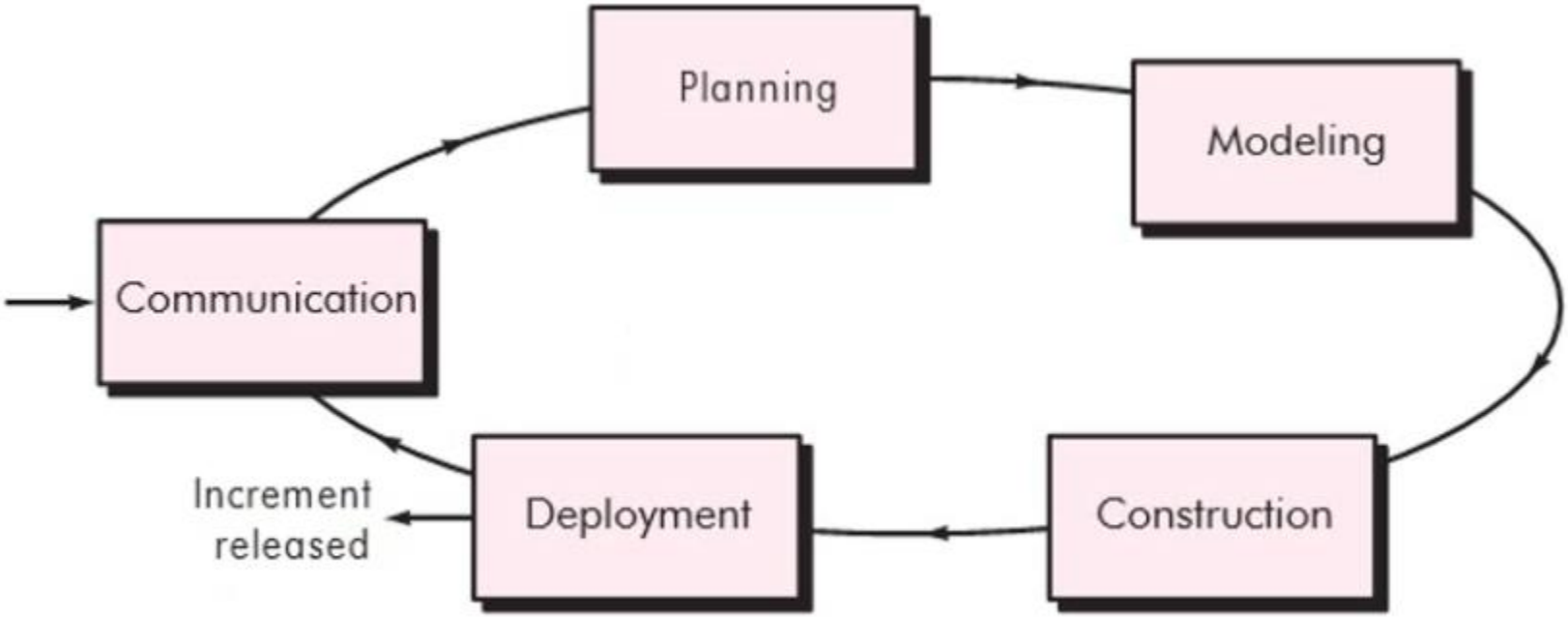


Iterative process flow

۳- جریان فرآیند تکاملی

در این جریان فعالیت ها به شیوه ی حلقوی اجرا می شوند .

هر مدار از ۵ فعالیت عبور می کند که به نسخه کاملتری از نرم افزار می انجامد .

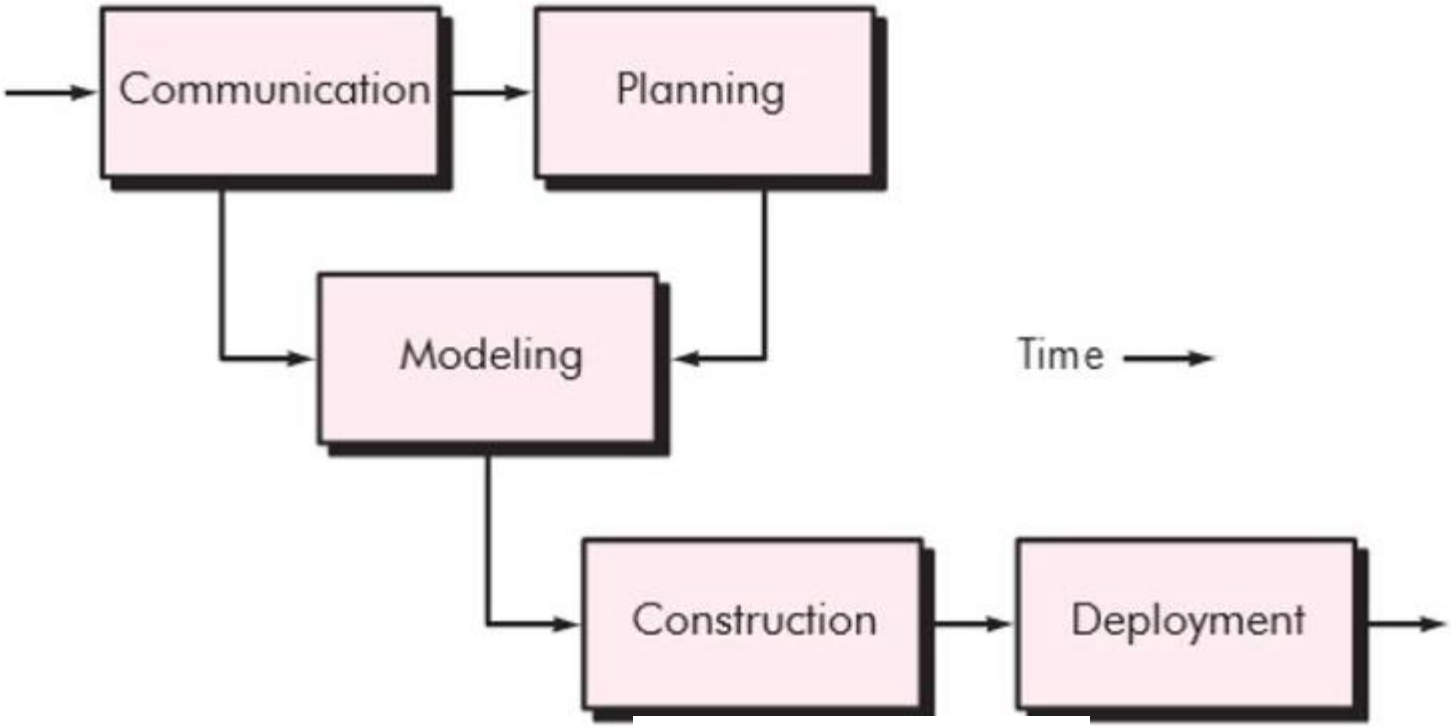


Evolutionary process flow

۴- جریان فرآیند موازی

یک یا چند فعالیت به موازات سایر فعالیت ها انجام می شود .

مثلاً مدل سازی برای یک جنبه از نرم افزار ممکن است به موازات ساختار جنبه ی دیگری از نرم افزار اجرا گردد.



Parallel process flow

انواع چرخه‌های حیات تولید و توسعه نرم افزار

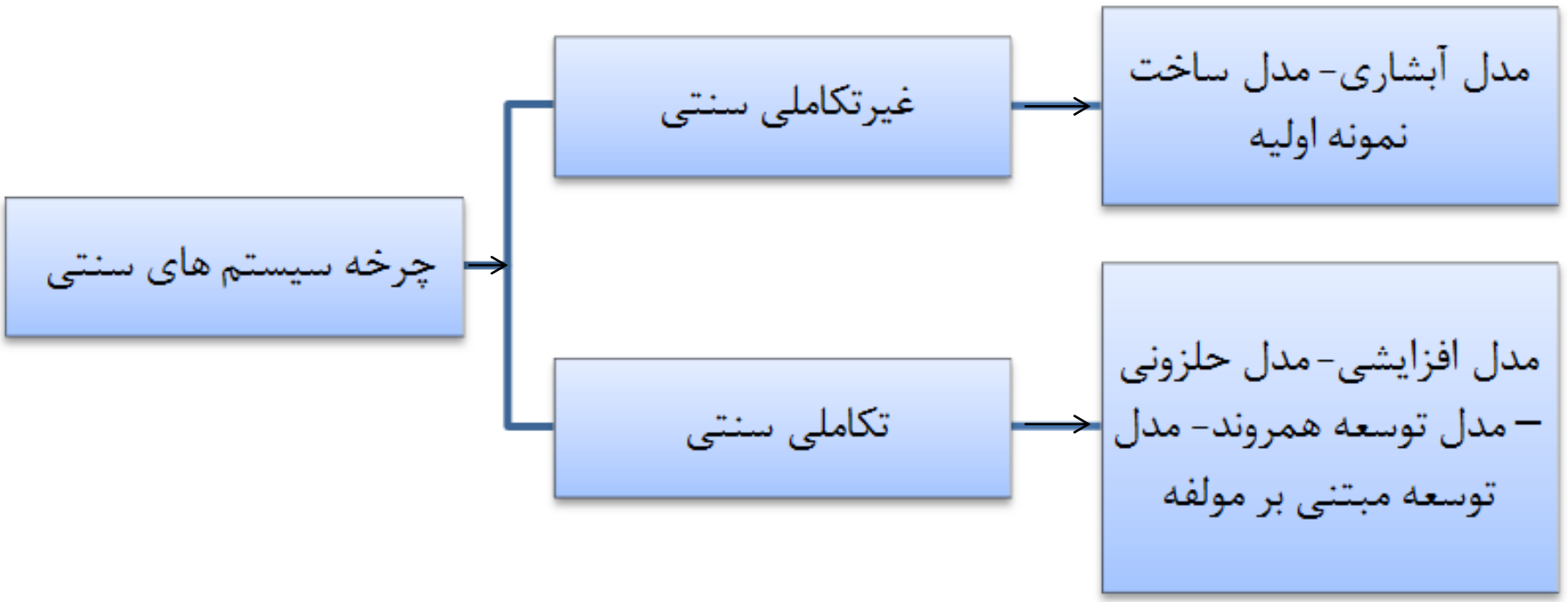
- مخفف System Development Life Cycle می‌باشد .
- ۵ فعالیت نامبرده که طی تولید و توسعه نرم افزار به کار می‌روند را چرخه حیات تولید و توسعه نرم افزار می نامند به دو دسته تقسیم می شود:

۱. چرخه حیات سیستم‌های سنتی(ساخت یافته) یا TLC
۲. چرخه حیات سیستم‌های مدرن (شی گرا) یا OODLC

۱-چرخه حیات سیستم‌های سنتی یاTLC

- TLCمخفف Traditional Life Cycle است .
- در گذشته چون برنامه ها به صورت فرایند گرا یا Process Orientedنوشته می‌شدند از روش TLC برای تولید و توسعه نرم افزار استفاده می شد .

دسته بندی مدلهای مورد استفاده در TLC(مدل های فرایند سنتی)



- قبل از برنامه نویسی شئ گرا عموماً نرم افزارها با روش برنامه نویسی ساخت یافته تهیه می شدند. این روش نمی توانست در مورد نرم افزارهای با حجم بالا بخوبی جوابگو باشد .
- زیرا با افزایش تعداد خطوط برنامه و افزایش تعداد برنامه نویسانی که روی یک پروژه نرم افزاری کار می کردند مدیریت پروژه ،کارمشکلی بود و در صورت وجود خطا در نرم افزار امکان مکان یابی آن دشوار بود.

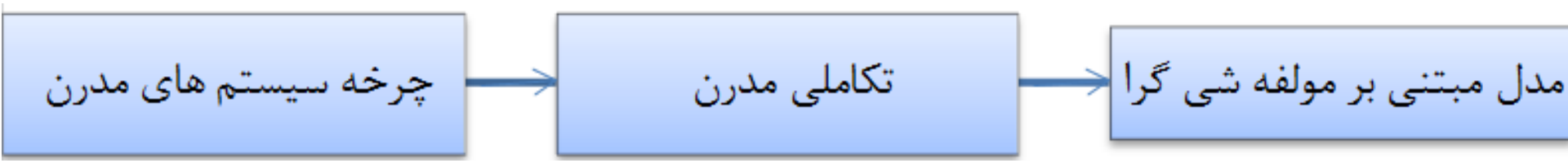
در نتیجه مدت زمان و هزینه اجرای پروژه ها رو به افزایش گذاشت .

از این رو شیوه برنامه نویسی شئ گرا مطرح شد.

۲-چرخه حیات سیستم های شی گرا یا OODLC

- این نوع چرخه حیات بعد از بوجود آمدن روش جدید برنامه نویسی یعنی روش شی گرا بوجود آمد .
- زبانهایی مانند C++, C#, java و بسیاری دیگر از زبانهای برنامه نویسی که قابلیت پیاده سازی خواص Object Oriented (شی گرایی) را دارا هستند امروزه از مهمترین زبانهای برنامه نویسی دنیا محسوب می شوند .

چرخه حیات سیستم‌های OODLC(مدل فرآیند مدرن)



مدل های فرآیند تجویزی(سنتی)

PRESCRIPTIVE PROCESS MODELS

این مدل در ابتدا برای نظم بخشیدن جهت توسعه نرم افزار ارائه شده است.

۱- مدل آبشاری — The Waterfall Model

۲- مدل های فرایند افزایشی — Incremental Process Models

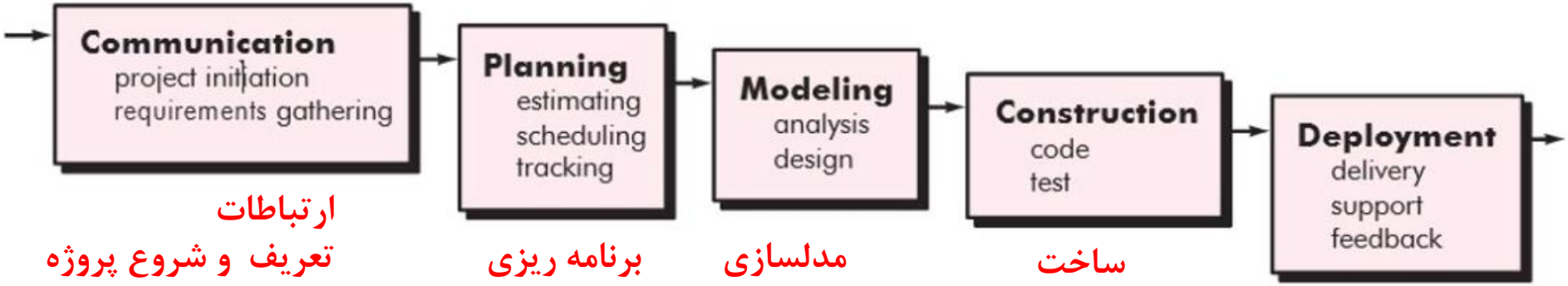
۳- مدل های فرایند تکاملی — Evolutionary Process Models

۴- مدل توسعه همروند — Concurrent Models

۱- مدل آبشاری

The Waterfall Model

در این مدل تا یک فعالیت به پایان نرسد فعالیت بعدی نمی تواند آغاز شود. خصوصیت اصلی این مدل این است که هیچ گونه بازخوردی بین فعالیت های این مدل وجود ندارد، مانند آب در آبشار که نمی تواند به عقب برگردد، در این مدل نیز بعد از ورود به یک فعالیت ،به فعالیت های قبلی نمی توان بازگشت.



ارتباطات
تعریف و شروع پروژه
جمع اوری نیازمندیها

برنامه ریزی
براورد
زمانبندی
پیگیری

مدلسازی
تحلیل
طراحی

ساخت
کدنویسی
تست

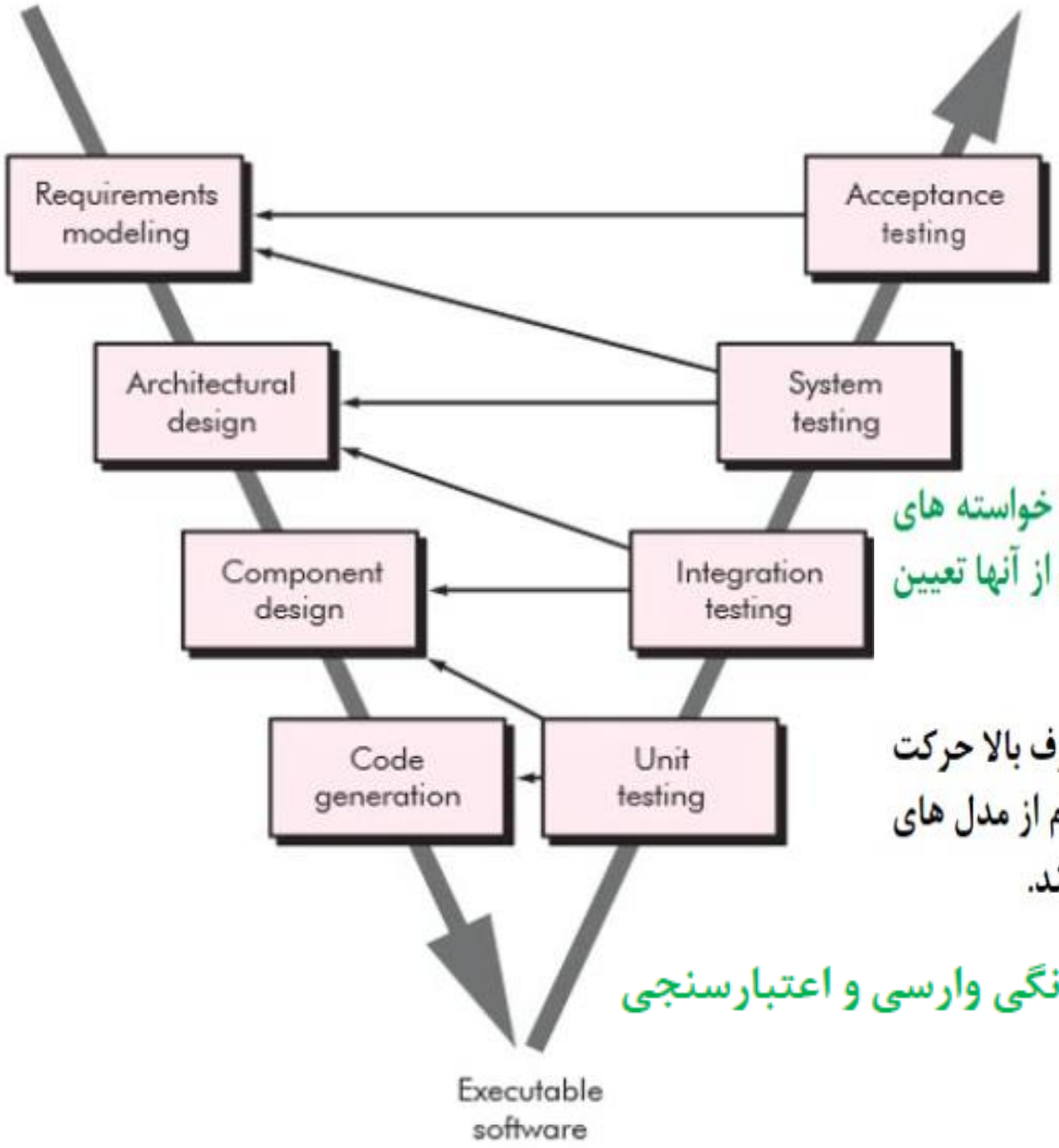
استقرار
تحویل محصول
پشتیبانی
فیدبک محصول

ویژگی های مدل آبشاری

- قدیمی ترین مدل در مهندسی نرم افزار می باشد.
  - بیان کامل و صریح نیازها در ابتدای کار برای مشتری دشوار است و در طول زمان این امر محقق می شود.
  - مدت زمان از جمع آوری نیازها تا ارائه خروجی به مشتری زیاد است.
  - با توجه به ترتیبی بودن کار، تعدادی از نیروی کار بیکار می مانند.
  - اعتبار هر فعالیت با نتایج و اهداف فعالیت قبل سنجیده شده و بررسی و تایید می‌شود.
- امروزه کار نرم افزار نیازمند سرعت است و تغییرات در آن پایان ناپذیر. مدل آبشاری برای این کار مناسب نیست.
- تنها در شرایطی مناسب است که نیازمندیها ثابت باشد.

➢ شکل دیگر در نمایش مدل آبشاری

مدل V

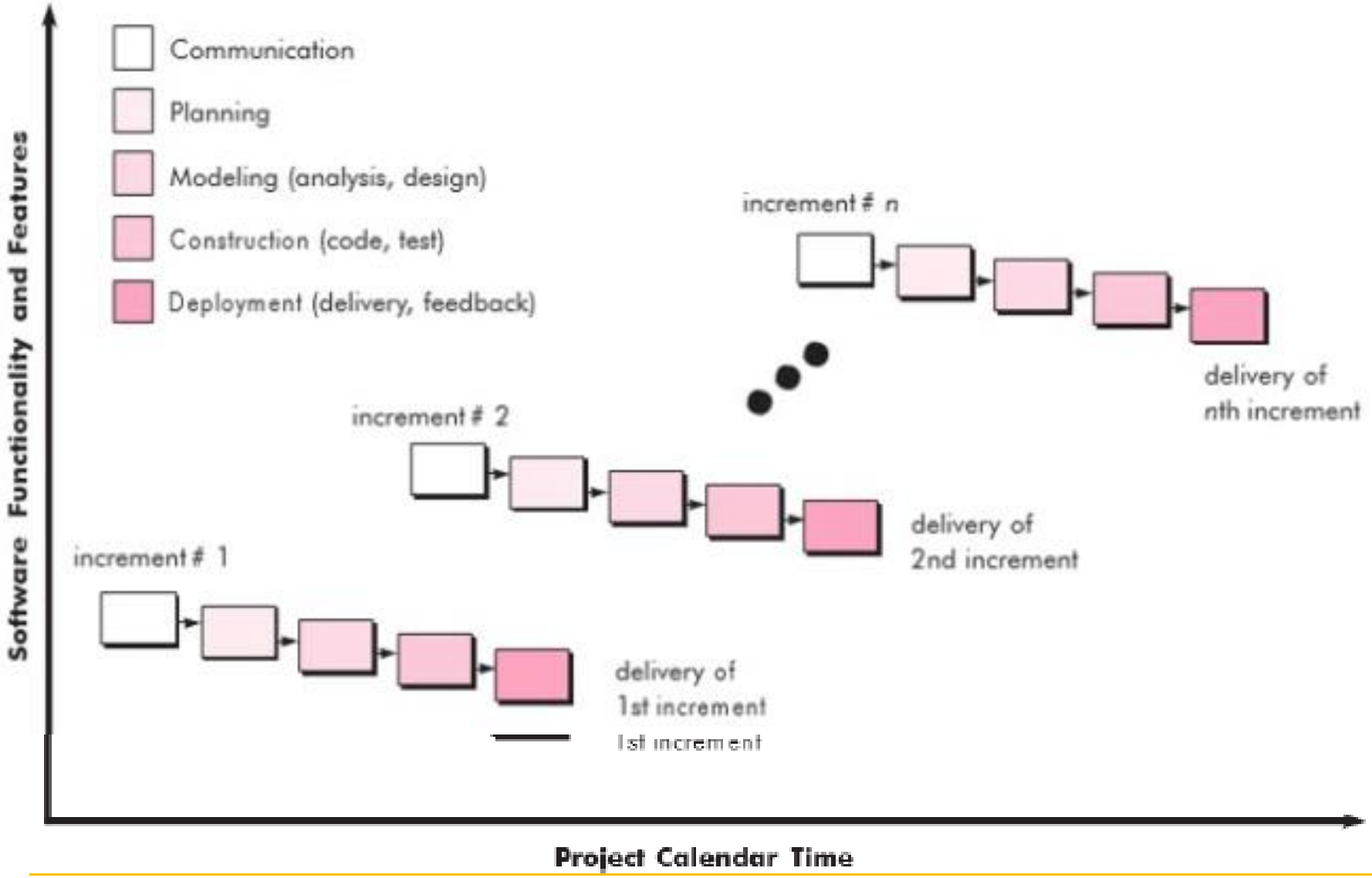


با حرکت تیم نرم افزاری به طرف پایین و سمت چپ V خواسته های اساسی مسئله رفته رفته پالایش شده و جزئیات بیشتری از آنها تعیین می شود و مسئله و راهکار آن بهتر نشان داده می شود .

هنگامی که کدها نوشته شد، تیم در طرف راست V به طرف بالا حرکت می کند و اساسا یک سری آزمون اجرا می کند تا هرکدام از مدل های ایجادشده در مدت حرکت تیم به طرف پایین را وارسی کند.

مدل V راهی برای تجسم بخشیدن به چگونگی وارسی و اعتبارسنجی در ابتدای کار نرم افزار را فراهم می آورد.

۲- مدل های فرایند افزایشی (Incremental process model)



مدل افزایشی

- در این مدل به تدریج پروژه تکمیل می شود یعنی هر مرحله که می گذرد پروژه کامل تر شده و در نسخه های بعدی این تکامل ادامه می یابد تا به هدف اصلی برسیم.
- درانتهای هر ترتیب خطی یک «رشد» قابل تحویل دهی را تولید می کند. اولین محصول با نام محصول هسته ای (Core Product) به نیازمندیهای پایه ای پرداخته و پس از بازنگری توسط کاربر اصلاح و بهینه می گردد.
- این مدل معمولا در مواقعی استفاده می شود که امکانات کافی در اختیار نداشته باشیم و مجبور باشیم از حداقل امکانات(نیروی انسانی و مالی و...) در مراحل مختلف برای ایجاد نتیجه ی مطلوب استفاده کنیم.

نمونه پروژه به روش افزایشی:

نرم افزار واژه پردازی که با استفاده از الگوی افزایشی توسعه یافته است، ممکن است اعمالی از قبیل:

مدیریت فایل، تولید و ویرایش مستندات در نسخه ی اول

قابلیتهای پیچیده تر ویرایشی و تولید مستندات در نسخه دوم

چک کردن املا و دستور در نسخه سوم

قابلیت های پیشرفته ی صفحه بندی را در نسخه چهارم

تحویل دهد.

۳- مدل های فرایند تکاملی

ساخت نمونه اولیه

الگوی ساخت نمونه اولیه با جمع آوری خواسته ها آغاز می شود.

مشتری و سازنده با هم ملاقات می کنند و اهداف کلی نرم افزار را تعیین می کنند. همه ی خواسته های معلوم را شناسایی می کند و زمینه هایی را مطرح می کند که تعریف بیشتر در آنها ضروری است. سپس یک طراحی سریع صورت می گیرد.

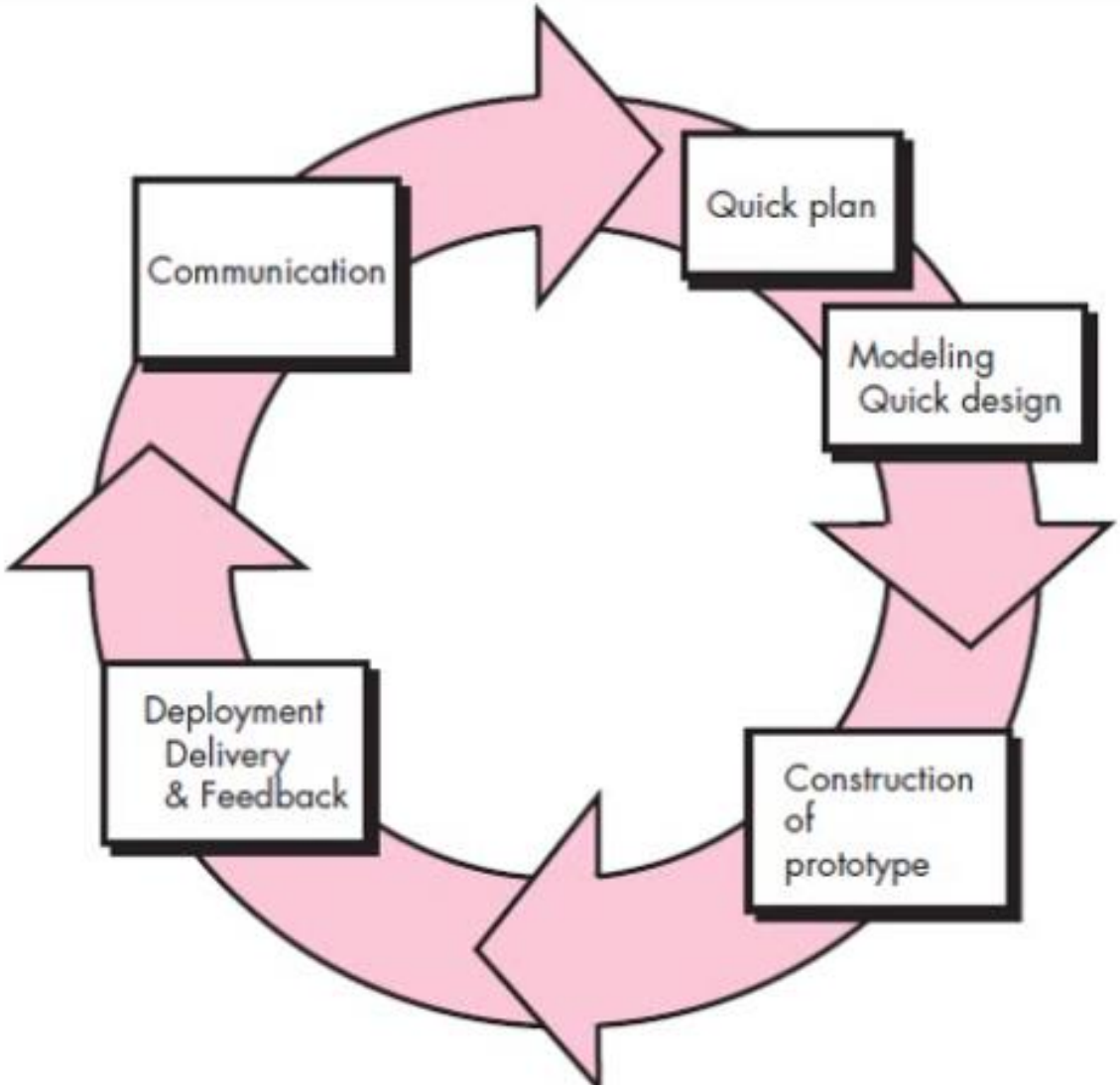
در طراحی سریع هدف اصلی ارائه آن دسته از ویژگی های نرم افزار است که مورد توجه کاربر می باشد مثل روش های واردکردن اطلاعات و فرمت های خارجی. طراحی سریع منجر به ساخت یک نمونه اولیه می شود.

نمونه اولیه مورد ارزیابی مشتری قرار گرفته و از آن برای پالایش نرم افزار مورد نظر استفاده می شود.

با تنظیم نمونه اولیه برای برآوردن نیازهای مشتری تکرار رخ می دهد و در عین حال سازنده بهتر می فهمد که چه نیازهایی باید برآورده شود.

مدل های فرایند تکاملی

ساخت نمونه اولیه



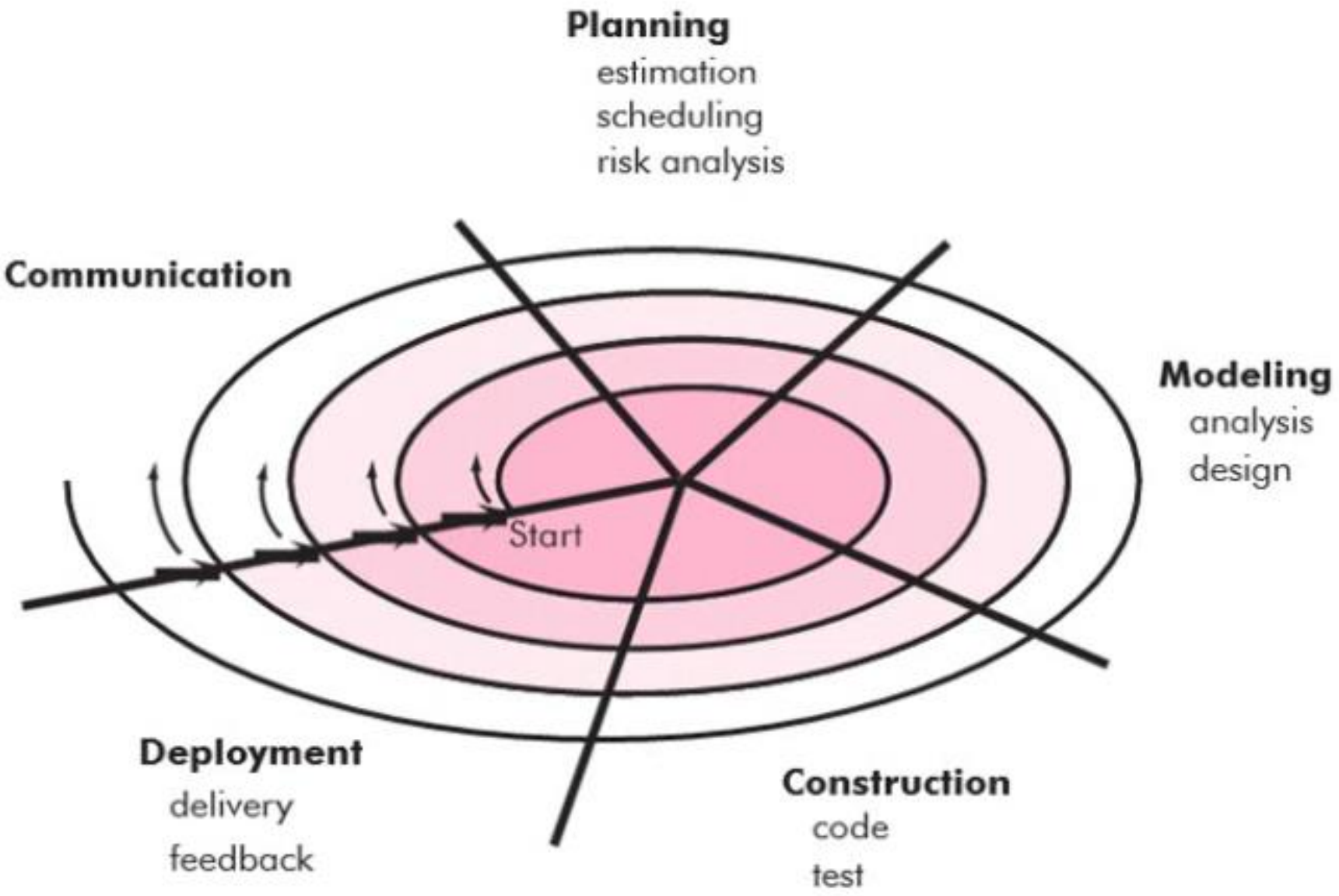
مدل های فرایند تکاملی

- مدلهای تکاملی، تکراری هستند. یعنی در هر تکرار، نسخه های کامل تر از نرم افزار تولید می شوند. نسخه های اولیه به عنوان نمونه اولیه برای ارزیابی توسط کاربر و درک اصلاحات بعدی به کار می رود.

- در این مدل ها در هر زمان اضافه کردن نیازهای جدید به نرم افزار با کمترین تغییرات، کمترین هزینه و بدون از دست دادن کیفیت نرم افزار انجام می گیرد.

۳.۱. مدل مارپیچی یا حلزونی (The spiral model)

- ✓ یک مدل فرایند تکاملی می باشد که از ترکیب مدل آبشاری و ساخت نمونه اولیه استفاده می شود بدین شکل که روش آبشاری است اما منطق استفاده شده همان منطقی است که در روش ساخت نمونه اولیه استفاده می شود.
- ✓ این مدل به تعدادی فعالیت تقسیم می شود که گاهی اوقات به آن ها نواحی کاری می گویند.


Planning
estimation
scheduling
risk analysis
Communication
Modeling
analysis
design
Start
Deployment
delivery
feedback
Construction
code
test

مدل حلزونی

- ✓ با شروع فرایند تکمیلی، تیم مهندسی نرم افزار در جهت حرکت عقربه های ساعت،حرکت در مارپیچ را آغاز می کند و این کار از مرکز شروع می شود. اولین مدار حول مارپیچ ممکن است منجر به یک مدل کاغذی یا یک نمونه اولیه شود. طی تکرارهای بعدی هر بار نسخه کاملتری از سیستم مهندسی شده تولید می شود.
- ✓ با عبور از هر مرحله منطقه برنامه ریزی، هزینه و زمانبندی بر اساس بازخورد ارزیابی مشتری تنظیم می گردند. علاوه بر آن ،مدیر پروژه ، تعداد تکرارهای تنظیم شده لازم برای تکمیل نرم افزار را تعیین می کند.
- ✓ در مدل حلزونی هر فلشی که در طول محور قرار داده شده است را می توان به عنوان نماینده نقطه شروع برای انواع پروژه های مختلف در نظر گرفت.

مدل حلزونی

- ✓ این مدل ،پتانسیل لازم برای بسط سریع نسخه های تکاملی نرم افزار را داراست.
- ✓ مدل حلزونی یک روش واقعی در توسعه سیستم ها و نرم افزارهایی است که دارای مقیاس بزرگ هستند.
- ✓ به دلیل اینکه نرم افزار به شکل تدریجی تکمیل می شود، توسعه دهنده و مشتری بهتر آن را درک می نماید و می تواند در مقابل سطح ریسک پذیری تکمیل پروژه واکنش نشان دهد.
- ✓ اگر یک ریسک عمده کشف و مدیریت نشود بدون شک مشکلاتی به بار می آورد.
- ✓ برخلاف سایر مدل های فرایند کلاسیک که با تحویل نرم افزار پایان می پذیرد، مدل مارپیچی را می توان طوری تطبیق داد که در سراسر عمر نرم افزار قابل استفاده باشد.

۴- مدل توسعه همروند

Concurrent Models

مدل توسعه هم روند که با نام مهندسی هم روند نیز شناخته می شود، جهت اجرای چند پروژه همزمان مورد استفاده قرار میگیرد که هر یک از فعالیت های چارچوبی فرایند تولید نرم افزار مربوط به هر پروژه می تواند در وضعیت های مختلفی قرار داشته باشند.

خصوصیات مدل توسعه هم روند:

- در دنیای واقعی، مدل توسعه ی هم روند را در تولید تمامی انواع نرم افزار ها می توان به کار برد و این مدل، تصویری درست از وضعیت جاری یک پروژه را نمایش می دهد.
- در مدل توسعه هم روند، به جای تعریف ترتیبی برای فعالیت های چارچوبی، شبکه ای از فعالیت های چارچوبی برای هر پروژه خواهیم داشت. به نحوی که رخداد های یک فعالیت روی وضعیت آن فعالیت و سایر فعالیت های دیگر تاثیر می گذارد.

مدل توسعه همروند

Concurrent Models

مدل توسعه همروند

Concurrent Models

شکل طرحی از یک فعالیت با مدل توسعه همروند ارائه می دهد . این فعالیت مدل سازی ممکن است در هر زمان و در حالت های مختلف ایجاد شود برای مثال در ابتدای یک پروژه فعالیت برقراری ارتباط نخستین تکرار خود را به پایان رسانده و در حالت انتظار تغییرات قرار دارد در واقع فعالیت مدل سازی که هنگام کامل شدن ارتباط اولیه با مشتری در حالت غیرفعال قرارداشت اکنون دستخوش گذار به حالت تحت توسعه می شود ولی اگر مشتری متذکر شود که تغییراتی در خواسته ها باید صورت گیرد فعالیت تحلیل از حالت تحت توسعه به حالت انتظار تغییرات می رود.

درواقع مدل توسعه همروند یک سری رویداد تعریف می کند که باعث گذار از حالتی به حالت دیگر برای هر یک از فعالیت های مهندسی نرم افزار می شود .

برای مثال طی اولین مراحل طراحی یکی از کنش های اصلی در مهندسی نرم افزار که طی فعالیت مدل سازی انجام می شود یک ناسازگاری در مدل تحلیل کشف می شود این باعث تولید رویداد تصحیح مدل تحلیل می شود که گذار از کنش تحلیل خواسته ها را از حالت انجام شده به حالت انتظار تغییرات سبب می شود.

مدل های فرایند تخصصی

SPECIALIZED PROCESS MODELS

این مدل ها شامل بسیاری از ویژگی های یک یا چند مدل سنتی ارائه شده در بخش های قبلی می شوند. ولی، این مدل ها را معمولا هنگامی به کار می برند که یک روش مهندسی تخصصی یا روشی با مشخصات دقیق انتخاب می شود.

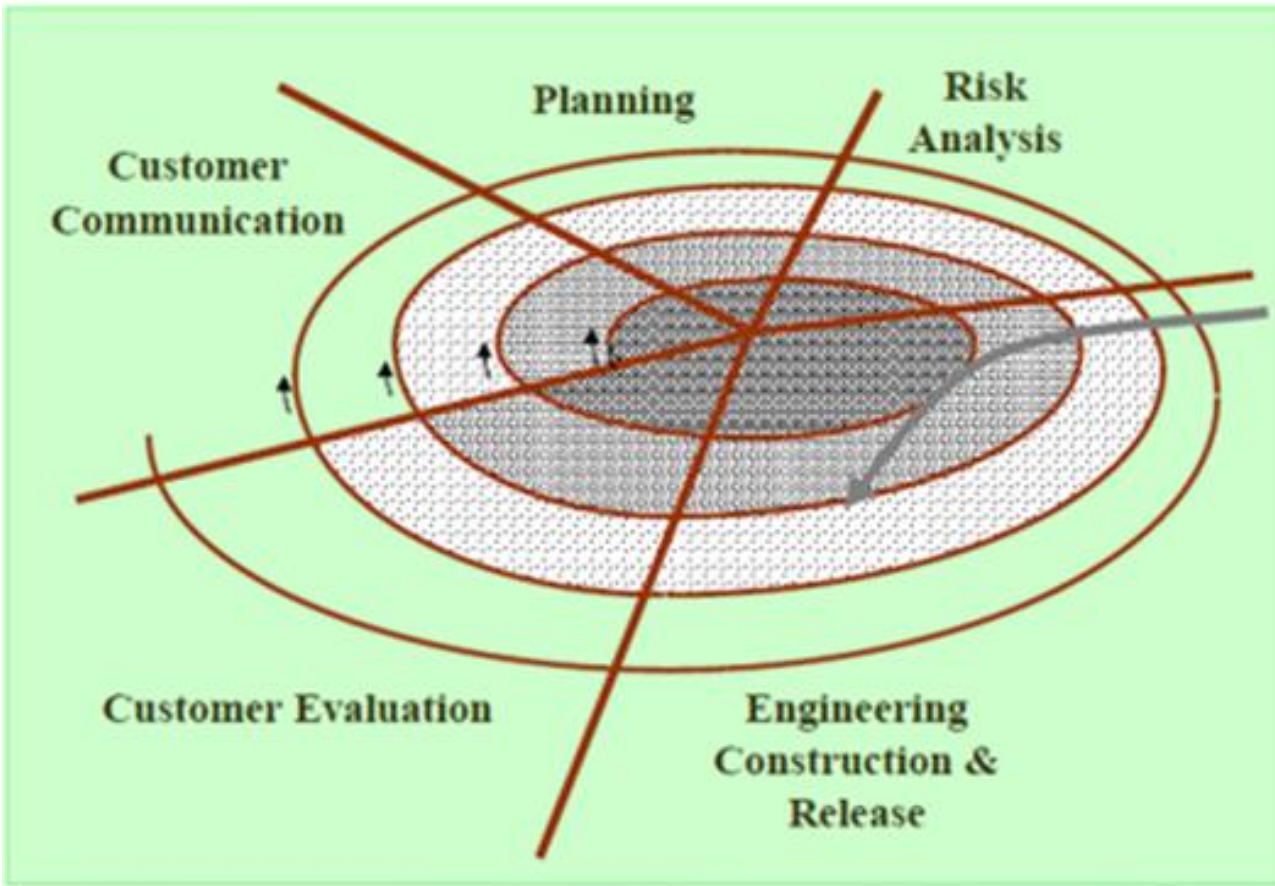
۱- توسعه مبتنی بر مولفه ها

۲- مدل روش های رسمی

۳- توسعه نرم افزار به روش جنبه گرا

۱- توسعه مبتنی بر مولفه
(Component Base Development)

مدل فرایند توسعه مبتنی بر مولفه ،برای طراحی نرم افزار به روش شی گرا(مدل مدرن) تدوین شده است.



مراحل مدل توسعه مبتنی بر مولفه

۱) شناسایی مؤلفه های مورد نیاز
۲) جستجوی مؤلفه ها در کتابخانه
۳) اگر مؤلفه موجود بود بازیابی و استفاده می شود.
۴) اگر مؤلفه مورد نظر وجود نداشت آن را می سازیم
۵) به مؤلفه های آماده در کتابخانه اضافه می کنیم.
۶) این مراحل تکرار می شود. نخستین نسخه نرم افزار با مونتاژ مؤلفه ها بدست می آید.

- ✓ مؤلفه های نرم افزاری آماده، توسط عده ای از فروشندگان این مؤلفه ها توسعه داده می شوند، عملکرد مورد نظر را با واسطه هایی مناسب فراهم می آورند، به طوری که مؤلفه را می توان به خوبی در سیستم در حال ساخت الحاق کرد.
- ✓ توسعه مبتنی بر مؤلفه بسیاری از خصوصیات مدل مارپیچی را در بر می گیرد.

مزایای مدل توسعه مبتنی بر مؤلفه

استفاده مجدد از مؤلفه ها بهترین مزیت بوده و خود مزایای زیر را به دنبال دارد:

- ✓کاهش در زمان چرخه توسعه نرم افزار
- ✓کاهش در هزینه پروژه
- ✓افزایش ضریب بهره وری

۲-مدل روش های رسمی(Formal Method)

مدل روش های رسمی (فرمال، قراردادی و صوری) شامل مجموعه ای از فعالیت هاست که سعی دارد پروژه ی نرم افزاری را در قالب روابطی رسمی و ریاضی، تعریف، توسعه، پیاده سازی و ارزیابی نماید.

در این مدل، با استفاده از تحلیل های ریاضی بسیاری از ابهامات، نواقص و عدم سازگاری نرم افزار را تا حد زیادی می توان به سادگی کشف و تصحیح نمود.

گونه ای از روش های رسمی وجود دارد که به مهندسی نرم افزار اتاق تمیز معروف است. این مدل، امکان کشف و تصحیح خطاهای زیادی را که تا زمان اجرا غیر قابل تشخیص هستند را در طول مراحل اولیه تولید نرم افزار، فراهم می کند.

مدل روش های رسمی(Formal Method)

کاربرد:

توسعه نرم افزارهای بحرانی و امنیتی مانند نرم افزارهای دستگاه های پزشکی و هوافضا.

توسعه نرم افزارهایی که بروز خطا در آنها موجب زیان های اقتصادی کلان می شود.

معایب:

بسیار پرهزینه و زمانبر می باشد.

- آموزش گسترده ای برای یادگیری روش های رسمی مورد نیاز است.
- استفاده از این روش برای ارتباط با مشتریانی که دید فنی ندارند دشوار است.

۳- توسعه ی نرم افزار به روش جنبه گرا

Aspect-Oriented Software Development(AOSD)

زمانی که می خواهیم نرم افزاری را با مدل جنبه گرا پیاده سازی کنیم در مرحله شناسایی نیازها و طراحی معماری باید این مساله را در نظر گرفت که دغدغه ها به عنوان یک اصل برای بدست آوردن نیازمندی ها استفاده شود و بنابراین، نیازهای سیستم را با دید جنبه گرایی بدست می آید. یکی از دغدغه های کاربران(مثل امنیت) می باشد باید نیازهای آنان را شناخت و سپس اقدام به جداسازی نیازهای همراه و هسته اصلی نمود .

به عنوان مثال در سیستم حفظ اطلاعات بیماران، هسته اصلی سیستم امکان ایجاد، ویرایش ، مدیریت و دسترسی به داده های بیماران در پایگاه داده است .

نمونه هایی از نیازهای همراه : وضعیت کنترل دسترسی کاربران به داده های پایگاه داده و یا رمز نگاری آن داده ها می باشد.

مدل های فرایند تیمی و شخصی

PERSONAL AND TEAM PROCESS MODELS

بهترین فرآیند نرم افزار ، فرآیندی است که به کسانی که کار می کنند نزدیک باشد.

فرآیند نرم افزار شخصی Personal Software Process (PSP)

فرایند نرم افزار تیمی Team Software Process (TSP)

فرآیند نرم افزار شخصی

Personal Software Process (PSP)

هر سازنده ای برای ساختن نرم افزار کامپیوتری از یک فرایند استفاده می کند.
psp بر اندازه گیری کیفیت محصول کاری و همچنین بر شناخت خطاها قبل از بروز تاکید دارد.
در صنعت از این روش زیاد استفاده نمی شود نیاز به آموزش طولانی دارد و هزینه بر می باشد.
از این روش به عنوان یک فرایند نرم افزار اثربخش در سطح شخصی می توان استفاده کرد.

فعالیت های چارچوبی در مدل PSP

۱) برنامه ریزی Planning :شناسایی خواسته ها- زمانبندی پروژه و....

۲) طراحی سطح بالا High-level design ساخت نمونه اولیه و...

۳) مرور طراحی سطح بالا High-level design review یافتن خطاهای زمان طراحی با استفاده از روش های وارسی رسمی

۴) توسعه Development پالایش و بازبینی طراحی سطح بالا -کدنویسی و تست

۵) پایان کار Postmortem

با استفاده از معیارهای جمع آوری شده اثربخشی فرایند تعیین می شود.

فرایند نرم افزار تیمی

Team Software Process (TSP)

هدف TSP تشکیل یک تیم پروژه خود هدایت گراست که سازماندهی برای تولید نرم افزارهای با کیفیت را خود عهده دار می شود. افراد باید از یک سطح همکاری داخلی و ارتباط خارجی عالی برخوردار باشند

فعالیت های زیر برای TSP تعریف می شود:

۱- تشکیل تیم های خود هدایت گری که کار خود را برنامه ریزی و پیگیری می کنند، اهداف را تعیین می کنند و خود به تعیین فرآیندها و طرح ها اقدام می نمایند. این تیم ها می توانند تیم های نرم افزاری محض یا تیم های محصولات انسجام یافته شامل ۳ تا حدود ۲۰ مهندس باشد.

۲- نشان دادن شیوه ی راهبری و ایجاد انگیزه در تیم ها به مدیران و چگونگی کمک به آنها در حفظ حداکثر کارایی

۳- شتاب بخشیدن به بهبود فرآیند نرم افزار با نهادینه ساختن CMM سطح ۵.

۴- فراهم ساختن دستور العمل بهسازی برای سازمان های بالغ

۵- تسهیل آموزش دانشگاهی مهارت های تیمی در سطح صنعتی.

یک مدل بلوغ شایستگی می باشد که میزان اثربخشی یک فرایند است.

چارچوب فعالیت ها در فرایند نرم افزار تیمی

۱- آغاز پروژه : شناسایی نیازمندیها-برنامه ریزی

۲- طراحی سطح بالا

۳- پیاده سازی نرم افزار به روش منظم ساخته می شود

۴- انسجام دهی: در حالی که اندازه گیری کمی و کیفی نرم افزار انجام می شود.

۵- آزمون: پایان کار-انجام عمل بهسازی فرآیند.

فرایند یکپارچه

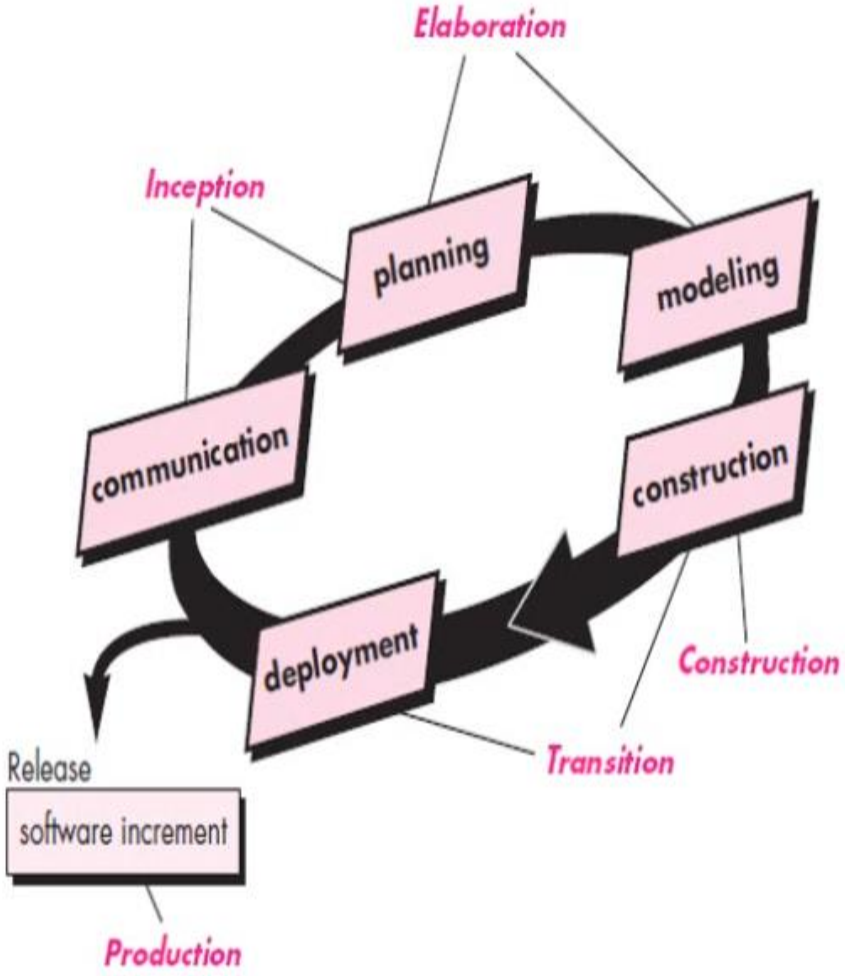
THE UNIFIED PROCESS

یک روش تکراری و افزایشی است برای توسعه‌ی نرم‌افزارهای شیءگراست که به عنوان روش استاندارد توسعه‌ی شیءگرا در کنار زبان UMLشناخته می‌شود .

این روش نسخه‌ی ساده‌تر و غیرتجاری روش بسیار معروف و تجاری فرایند جامع رشنال یا همان RUPاست. (البته RUP فراتر از یک روش است، بلکه یک چارچوب فرایند است که برای سازمان‌ها و پروژه‌ها استفاده می شود)

مراحل فرآیند یکپارچه

Phases of the Unified Process



مرحله اول : شامل هر ۲ فعالیت برقراری ارتباط با مشتریان و برنامه ریزی می شود.

مرحله شناخت: شامل فعالیت های برنامه ریزی و مدل سازی در مدل فرآیند کلی می شود.

در این مرحله یوزکیس های مقدماتی که به عنوان بخشی از مرحله آغازین ایجاد شوند، پالایش یافته و بسط داده می شوند.

مرحله ساخت : هم ارز فعالیت ساخت است که برای فرایند نرم افزار کلی تعریف شد.

مرحله گذار: شامل آخرین مراحل در فعالیت ساخت در مدار کلی و اولین بخش از استقرار در مدل کلی می شود.

مرحله تولید: منطبق بر فعالیت استقرار در فرآیند کلی است.

چگونگی آغاز یک پروژه

در اینجا مطالب این فصل به صورت کاربردی با شروع یک پروژه توضیح داده شده است .

این پروژه توسط شرکتی فرضی به نام CPI تولید می شود اتاق کنفرانس این شرکت در هنگام شروع پروژه شامل ۴ نفر می باشد. که مکالمات بین افراد جلسه بررسی خواهد شد تا با جلساتی که در شرکت جهت ایجاد پروژه تشکیل می شود آشنا شوید.

نام پروژه: Safe Home

تعریف پروژه : می خواهیم یک دستگاه بی سیم بسازیم که لوازم خانه را از طریق یک وب سایت یا کامپیوتر از بیرون کنترل کنیم. مثلا با ماشین تو راه منزل هستیم کولر یا چراغ منزل را روشن کنیم یا مثلا تصاویر دوربین های منزل را کنترل کنیم و ...

چگونگی آغاز یک پروژه

مدیر ارشد

مدیر بازرگانی

مدیر مهندسی

معاون مدیر اجرایی

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

مکالمات بین افراد جلسه

معاون مدیر اجرایی به مدیر مهندسی می گوید: شنیدم افراد زیر نظر شما در حال ساخت یک جعبه بی سیم جهانی هستند.

مدیر مهندسی: بله . چیز خیلی خوبی هست به اندازه یک قوطی کبریت است که می توان به هر سنسوری وصل کنیم حتی به دوربین های دیجیتال و با یک پروتکل بی سیم کار می کند.

معاون مدیر اجرایی به مدیر ارشد می گوید: شما موافق انجام این پروژه هستید.

مدیر ارشد: موافقم با فروش بدون رشدی که امسال داشتیم این پروژه خوبی هست.

من و مدیر بازرگانی تحقیقاتی در بازار انجام دادیم فکر می کنیم به یک خط جدیدی از محصول دسترسی داشته باشیم که خیلی خوب هست.

مدیر ارشد به مدیر بازرگانی می گوید: ایده ات را بگو.

مدیر بازرگانی: این محصول یک نسل کاملا جدید است اسمش گذاشتیم safe home یا محصولات مدیریت خانگی. این محصول از یک رابط بی سیم استفاده می کند و سیستمی در اختیار مالکان قرار می دهد که توسط کامپیوترهای شخصی می توان لوازم منزل راکنترل کرد. مثلا خاموش و روشن کردن وسایل منزل و ....

مدیر مهندسی: بخش مهندسی(افراد زیر نظر من) مطالعه امکان سنجی پروژه را انجام دادند. سخت افزارش می توان آماده کرد و هزینه اش کم هست . بخش نرم افزارش مهم هست ولی ما از عهده اش بر میایم.

مدیرارشد: بیشتر خانه ها کامپیوتر دارند اگر قیمت خوبی روی محصول بگذاریم خیلی عالی می شود و ما از شرکت های دیگر خیلی جلو می افتیم و یک پرش دو ساله داریم و سال دوم می توانیم تا ۴۰ میلیون دلار سود کنیم.

ادامه پروژه Safe home–انتخاب یک مدل فرایند(بخش ۲)

SafeHome

Selecting a Process Model, Part 2

The scene: Meeting room for the software engineering group at CPI Corporation, a company that makes consumer products for home and commercial use.

The players: Lee Warren, engineering manager; Doug Miller, software engineering manager; Vinod and Jamie, members of the software engineering team.

ادامه پروژه SafeHome-مکالمات بین افراد جلسه

برای انتخاب یک مدل فرایند جهت تولید نرم افزار به صورت مهندسی با اعضای دیگر،جلسه دوم برگزار می شود

مدیر مهندسی: در راجع به خط تولید safe home وقت زیادی صرف کردم. برای اینکه تعریف ساده ای از پروژه ارائه دهیم باید خیلی کار کنیم و می خواهم بهم کمک کنید.

مدیر مهندسی نرم افزار : در گذشته در نگاه به نرم افزار خیلی منظم نبودیم.

عضو ۳: اما همیشه محصول را تولید کردیم.

مدیر مهندسی نرم افزار : درسته، ولی خیلی اذیت شدیم چون کار مهندسی روی نرم افزار انجام نمی دادیم. این پروژه هم بزرگتر از پروژه های قبلی می باشد.

عضو۱: زیاد سخت نیست. روشی که در پروژه های قبلی استفاده کردیم اینجا به کار نمیاد مخصوصا اگر از نظر زمانی محدودیت داشته باشیم.

مدیر مهندسی نرم افزار: هفته قبل دوره مهندسی نرم افزار شرکت کردم و مطالب جالبی یاد گرفتم. چارچوب فرایند و مدل فرایند تجویزی را توضیح می دهد ومی گوید اینجا یک فرایند لازم داریم و به نظر من مدل خطی جالب نیست چون در این پروژه از ابتدا نیازمندیها ثابت و مشخص نیست.

عضو۲: بله. مدل خطی بیشتر به درد پروژه های کنترل موجودی می خورد.

ادامه پروژه SafeHome-مکالمات بین افراد جلسه

عضو۳: روش نمونه سازی اولیه مناسب هست.

عضو۲: ولی این روش مشکل سازهست شاید ساخت یافتگی کافی در اختیار ما قرار ندهد.

مدیر مهندسی نرم افزار: گزینه های فرایند تکاملی را توضیح می دهد.

عضو۱: روش مارپیچی معقول هست چون واقعی به نظر میاد.

عضو۲: یک نسخه به مشتری می دهیم و هر بار نسخه جدید با قابلیت های جدید طبق بازخورد مشتری تولید می کنیم.

مدیر مهندسی: یعنی باید در هر چرخه تحویل نسخه جدید ، برنامه ریزی مجدد انجام شود؟ این کار جالبی نیست.

مدیر مهندسی نرم افزار: با هر بار تغییر برنامه ریزی، چیزهای بیشتری به دست می آوریم و واقع بینانه تر هست.