



عنوان: Degree Overview

زیر نظر: دکتر سامی

پدید آورندگان: مریم صباحی، فاطمه رحیمی

بهار 95

فهرست

3	Project Overview
3	هدف از پروژه
3	روند کاری
3	Normal requirements
4	Expected requirements
4	Exciting requirements
4	Project Deliverables
5	معرفی مدل ها
18	مدل انتخابی
18	دلیل انتخاب مدل
19	توضیح فاز های مدل بر روی پروژه
21	ERD
21	DFD
24	Function Hierarchy Diagram (FHD)
25	اعضای گروه
25	Recourses

Project Overview

این پروژه به منظور متوجه شدن و به تصویر کشیدن روابط بین COURSE ها، اهداف آموزشی و مباحثی که دانشجویان در انتهای ترم یاد میگیرند، طراحی شده است.

مثلا اگر در دانشگاه درس socket programming ارائه شود که پیش نیاز این درس دانسته هایی از جاوا است، اما در این دانشگاه نه جاوا و نه حتی درسی که به صورت مقدماتی برنامه نویسی را آموزش داده باشد، وجود نداشته باشد. اینچنین عدم تطابق های کوچک و بزرگی در این مورد بسیار اتفاق می افتد که در این پروژه تلاش برای حل اینچنین مشکلاتی را داریم.

هدف از پروژه:

هدف استفاده از DegreeOverview این است که این امکان را به اساتید و دانشجویان بدهد که ترتیب دروس در طول گرفتن مدرک را که به خوبی به تصویر کشیده شده ، مشاهده نمایند. و همچنین به این سؤال دانشجویان که در طول تحصیل خود چه چیزی یاد گرفته اند، پاسخ داده شود.

روند کاری:

مدیر فنی سایت مسئولیت ایجاد دانشکده، رشته و گرایش و ثبت نام اساتید، دانشجویان و کارشناسان بخش در سیستم و همچنین ویرایش و حذف را به عهده دارد.

کارشناس بخش مسئولیت بررسی و تایید یا رد دروس درخواست داده شده توسط استاد و نظارت بر روی چارت درسی را برعهده دارد.

استاد درس اطلاعات مربوطه را وارد و درخواست ثبت درس را میدهد. در طول ترم بر عملکرد دانشجویان نظارت دارد و عملکرد آن ها را ارزیابی و ثبت میکند.

دانشجو با توجه به چارت و دروسی که ارائه میشود، درس دلخواه خود را بررسی و اخذ می کند. همچنین قادر به مشاهده عملکرد تحصیلی خود می باشد.

Normal requirements

✓ برای ارائه دهنده درس:

- به هنگام ثبت درس، معیاری (به عنوان سطح سختی درس) را معرفی میکند که دانشجو با این معیار میتواند عملکرد خود را در این درس بسنجد.
- به تصویر کشیدن ارتباطات و وابستگی های هر درس در دوره تحصیل و گرفتن مدرک (چارت). به این شکل که در هر درس مطالبی که با گذراندن آن دانشجو یاد میگیرد و میتواند انجام دهد، ذکر می شود.
- دیدن نمرات یک درس و تمام دروس دیگر دانشجویان در طی گرفتن مدرک.
- جستجو یک مبحث در دروس ارائه شده که بتواند پیش نیاز دروس دیگر باشد را مشخص کند (پیش نیاز ها و پس نیاز ها را مشخص کند). و همچنین جست و جو در دروس که چه درسی مبحث خاص جستجو شده را آموزش می دهد.

✓ برای دانشجویان:

- به تصویر کشیدن ارتباطات و وابستگی های هر درس در دوره تحصیل و گرفتن مدرک (چارت). به این شکل که در هر درس مطالبی که با گذراندن آن دانشجو یاد میگیرد و میتواند انجام دهد، ذکر می شود.
- دیدن عملکرد خود در دروس گرفته شده.
- جستجو یک مبحث در دروس ارائه شده که بتواند پیش نیاز دروس دیگر باشد را مشخص کند (پیش نیاز ها و پس نیاز ها را مشخص کند). جست و جو در دروس که چه درسی مبحث خاص جستجو شده را یاد می دهد.

:Expected requirements

وارد کردن ارزیابی ها (نمرات) توسط ارائه دهنده درس.

امکان اخذ درس با توجه به چارت.

وجود مدیر فنی سیستم برای افزودن دانشکده، رشته، گرایش، دانشجو و استاد و ایجاد هر تغییر در هر مورد.

نیاز به کارشناس بخش برای تایید دروس و نظارت بر چارت بخش مربوطه.

امکان ویرایش اطلاعات کاربری توسط کاربران.

:Exciting requirements

ارائه نمودار پیشرفت تحصیلی دانشجو در طی گرفتن مدرک.

مشاهده اطلاعات دانشجویان و اساتید توسط کارشناس بخش مربوطه.

:Project Deliverables

یک document کامل شامل overview ای از پروژه، اهداف این پروژه، روند کاری، requirement ها، توضیحی از انواع مدل ها و مدل انتخابی ما، توضیح فاز های مدل انتخابی روی پروژه، نمودارهای DFD، ERD و FHD.

و همچنین در ابتدا یک prototype از UI سیستم ارائه میشود.

و سیستمی که ارائه میشود به این ترتیب میباشد:

یک مدیر فنی در سیستم وجود دارد که username و password او را به دانشگاه تحویل میدهیم. وظیفه ایجاد و ویرایش دانشکده، رشته، گرایش را به عهده دارد. ثبت نام و ویرایش دانشجویان و اساتید و کارشناسان بخش نیز بر عهده اوست.

کارشناس بخش وظیفه تأیید درس هایی که استاد درخواست ارائه میدهد را دارد و بر چارت نظارت میکند. و همچنین میتواند اساتید و دانشجویان بخش خود و چارت رشته های مختلف را مشاهده کند.

ارائه دهنده درس در این سیستم فرم درخواست ارائه درس را پر میکند و درخواست او به کارشناس بخش فرستاده می شود و پس از تایید درس به لیست دروس ارائه شده در ترم و چارت اضافه می شود. همچنین استاد امکان مشاهده دروسی که ارائه داده است، را دارد و دانشجویانی که آن درس را انتخاب کرده اند میتوانند ببینند و نمراتشان را وارد کند. و همین طور استاد میتواند عملکرد هر یک از دانشجویان را در طول دوره تحصیلی ببیند. استاد امکان جستجو در چارت ها و دروس و امکان مشاهده اطلاعات آن ها را دارد.

دانشجویان میتوانند دروسی که تا الان گرفته اند و نمراتشان را مشاهده کنند. همچنین امکان دیدن نمودار پیشرفت تحصیلی خود را دارند و امکان جست و جو و ثبت نام در درس ها را دارند. و همین طور دسترسی دیدن چارت و جست و جو در آن را دارند.

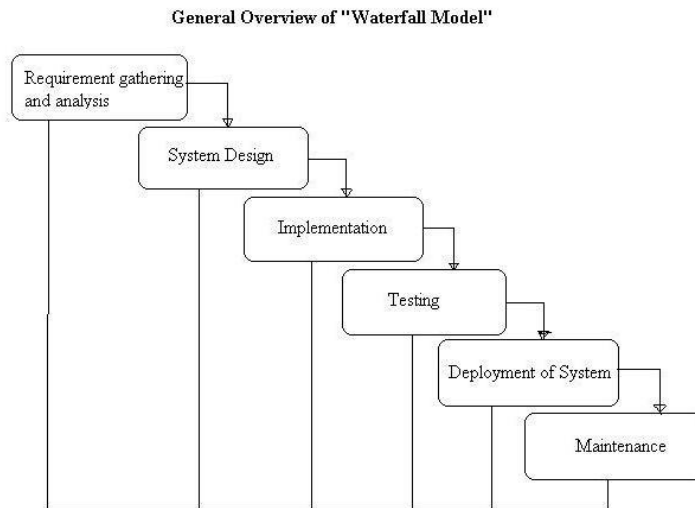
کارشناسان، اساتید و دانشجویان میتوانند اطلاعات کاربری خود را ویرایش کنند.

معرفی مدل ها:

🚧 مدل آبشاری (waterfall):

مدل آبشاری یک مدل ترتیبی توسعه و تولید نرم افزار است و در آن مراحل تولید به شکل یک جریان مداوم متمایل به سمت پایین است (همانند یک آبشار) که شامل فازهای تحلیل خواسته ها، طراحی، پیاده سازی یا implementation، آزمون و تست کردن، یکپارچه سازی یا integration، و دادن محصول به بازار می شود. مدیریت و مراحل تکمیل پروژه در این متدولوژی به سادگی قابل پیاده سازی است. زیرا در مرحله اول که مرحله بررسی نیازمندیهای پروژه می باشد، مشتری و تیم برنامه نویسی طی چند جلسه به بررسی نیازمندیها و خواسته های پروژه می پردازند. سپس پس از آن نوبت به مرحله طراحی می رسد، در مرحله ی طراحی افراد طرح کلی پروژه را می ریزند و

جزئیات پیاده‌سازی مشخص می‌شود. پس از مرحله‌ی طراحی تیم برنامه‌نویسی خود را برای پیاده‌سازی آماده می‌کند. در این مرحله همه قسمت‌های کد، پیاده‌سازی می‌شوند، در انتهای این مرحله، ما مرحله یکپارچه‌سازی یا integration را خواهیم داشت یکی از مشکلترین قسمت‌های انجام پروژه‌ها در این مرحله می‌باشد زیرا تنوع و گسترده‌گی کار کامل در این مرحله نقش دارد، هر چه میزان گسترده‌گی کار بالاتر باشد سختی یکپارچه‌سازی نیز بیشتر خواهد بود.



مزیت آشناری:

- 1- فهم این مدل ساده است.
- 2- از نظر تولید مستندات بهتر هستند.
- 3- مراحل قابل کنترل و بررسی هستند.

معایب آشناری:

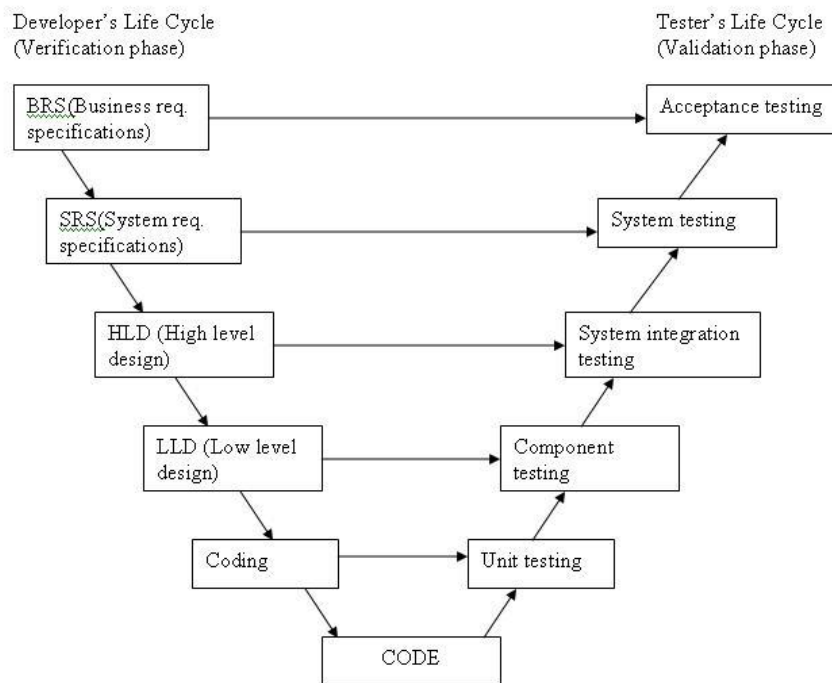
- 1- کار تیم‌های پایانی موقوف به اتمام کار تیم‌های ابتدایی می‌باشد.
- 2- بررسی بازخور سیستم توسط کاربران نهایی دارای پروسه مجزا و در انتهای کار میباشد.
- 3- گذر از مراحل بصورت نوبتی و وقت‌گیر میباشد.

مدل V:

مدل وی نماد یک روش فرآیند تولید نرم افزار است (که قابل تطبیق برای تولید سخت افزار هم میباشد) و ممکن است مدل آبشاری توسعه یافته به حساب آید. در این مدل به جای اینکه مسیر تولید همانند مدل آبشاری، یک مسیر شیب دار مستقیم به سمت پایین باشد، مسیر فرآیند ها پس از فاز برنامه نویسی به سمت بالا به شکل حرف وی انگلیسی خم می شود. مدل وی ارتباط بین فازهای مختلف چرخه حیات تولید نرم افزار و مراحل پیوسته فاز تست را مشخص می کند.

محورهای افقی و عمودی (از چپ به راست) میزان زمان یا تکمیل پروژه و سطح تکمیل مراحل که با عناوین انتزاعی تعریف شده اند (بالاترین سطح مفاهیم اصلی) را بترتیب نشان می دهد.

مدل وی در مقابل مدل آبشاری برای رسیدگی به توازن تولید نرم افزار و تایید پروژه استفاده می شود.



مزایا:

مشخصه خاص مدل وی در ایده های بررسی تایید و اعتبارسنجی آن است.

۱- کاربران در این مدل در تمام مراحل تولید، توسعه و نگهداری سیستم همراه هستند و فرم وضعیت تغییرات و نگهداری سیستم بصورت عمومی در انظار عمومی است. فرم وضعیت تمام درخواستها و تغییرات در طول سال را نشان می دهد.

۲- در ابتدای هر پروژه این امکان وجود دارد که، آن پروژه را بصورت یک مدل خاصی از وی مدل متناسب با آن پروژه طراحی کرد. چون مدل وی یک مدل سازمانی و مستقلی است.

۳- مدل وی کمک های قدرتمندی را در مورد چگونگی نحوه پیاده سازی فعالیت ها و مراحل زیر مجموعه هریک از کار، تعریف اتفاقات مورد نیاز جهت تکمیل کار را تشریح کرده است.

معایب:

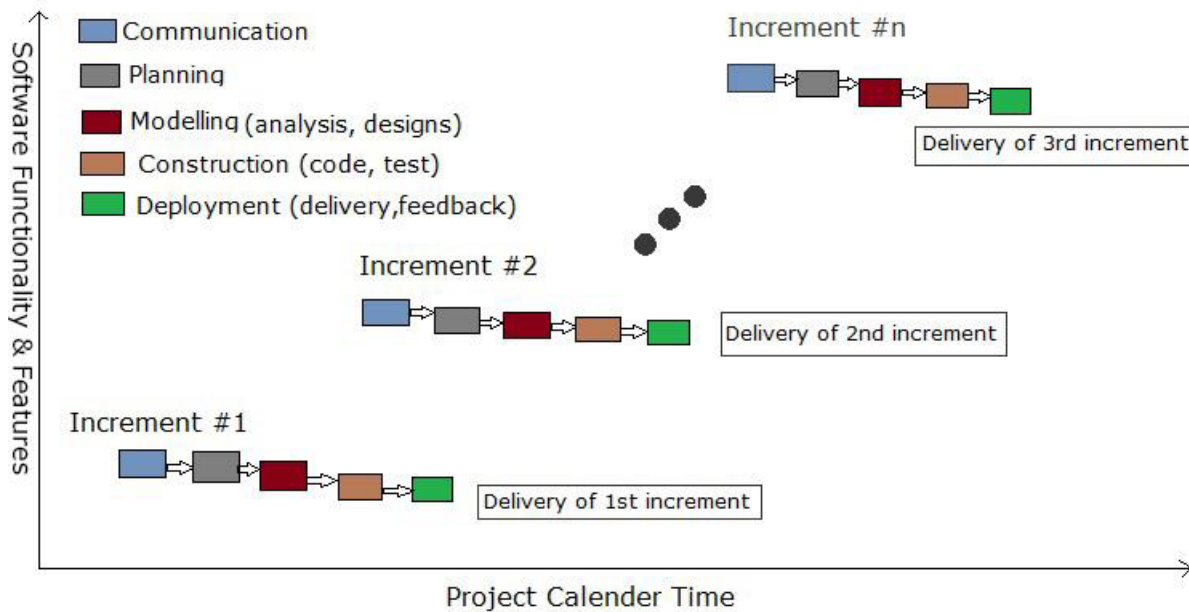
۱- برای کل پروژه با مدل وی نمی توان قرار دادی مشخص و شفاف مطرح کرد، و فقط برای زیر مجموعه های شفاف شده قرار داد معین نمود.

۲- در طول روال و مدت زمان مقدمه و نگهدارسیستم یک سازمان خاص فرقی بین عرضه کننده محصول و درخواست کننده نیست .

۳- سازماندهی و اجرای عملیاتها و تعمیر و نگهداری سیستم توسط این مدل پوشش داده نمی شود. با این حال برای برنامه ریزی و آماده سازی مستندات یک سیستم از این مدل استفاده می شود.

🚧 مدل افزایشی (incremental) :

در مدل افزایشی پروژه را به قسمت های کوچک تقسیم کرده و هربار قسمت مورد نظر کامل انجام و تست می شود. سپس قسمت بعد انجام می شود. هر یک از این قسمت ها خود از مدل آبشاری تبعیت می کند. از این مدل در پروژه هایی استفاده میشود که لازم است سریعاً به بازار ارایه شوند یا از تکنولوژی جدید استفاده می کنند مثلاً نرم افزار های موبایلی که از ایده ای نو بهره مندند و ویژگی های جدید به تدریج به آن افزوده شود.



مزایا:

- در این مدل پروژه به تدریج تکمیل می شود.
- در هر ترتیب خطی یک «رشد» قابل تحویل دهی را تولید می کند.
- در اینجا برخلاف روش نمونه ساز، نمونه های تولیدی مجدداً برای رشد بعدی قابل استفاده هستند.
- همانند روش نمونه ساز و سایر روش های افزایشی طبیعی تکراری دارد.
- برخلاف مدل نمونه ساز، در هر مرحله از رشد خود تمرکز روی تحویل یک محصول عملیاتی دارد.
- این مدل زمانی مفید است که از نظر نیروی انسانی و مالی امکانات کافی برای رسیدن به نتیجه مورد نظر در دسترس نباشد.
- عین روش نمونه سازی نسخه ها دور ریخته نمی شود.
- نرم افزار update می گردد به طوری که هزینه کمتری در بر داشته باشد.

معایب:

- هزینه آن از مدل آبشاری بیشتر است.
- نیاز به نقشه و طراحی خوب و دقیق دارد.
- لازم است قبل از تقسیم کردن پروژه، یک تعریف کامل و دقیق از تمام پروژه داشته باشیم.
- وقتی استفاده میشود که نیروی انسانی متخصص و یا سخت افزار کافی در اختیار نباشد.

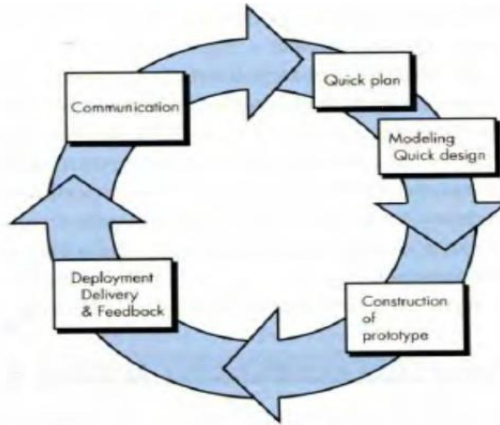
مدل prototyping :

اغلب مشتری یک سری اهداف کلی برای نرم افزار تعریف میکند اما ورودی ها پردازش یا نیاز های خروجی را به طور دقیق بیان نمیکند. در سایر موارد ممکن است تولید کننده از کارایی الگوریتم قابلیت تطابق آن با یک سیستم عامل یا شکل ارتباط متقابل دستگاه و انسان نامطمئن باشد. در این موارد و بسیاری از موارد دیگر ممکن است prototyping بهترین روش باشد.

Prototyping نخست با جمع آوری نیازمندی ها شروع میشود. مشتری و تولید کننده همدیگر را ملاقات میکنند و اهداف کلی خود را در مورد نرم افزار بیان می نمایند، هر چه که مورد نیاز هست را شناسایی می نمایند و حوزه هایی را مطرح می کنند که در آن تعریف بیشتر ضروری است. سپس طراحی سریع رخ میدهد. این طراحی روی نمایش آن جنبه هایی از نرم افزار متمرکز میشود که برای مشتری یا کاربر مشهود هستند. این طراحی سریع منجر به ساخت یک نمونه میشود.

نمونه نخست توسط مشتری یا کاربر ارزیابی شده و از آن برای رفع نیاز های نرم افزاری که قرار است تولید شود استفاده میشود. وقتی که نمونه اولیه طوری تنظیم شد که نیاز های مشتری را برآورده سازد، عمل تکرار رخ میدهد، در حالی که

در همان زمان تولید کننده نرم افزار را قادر میسازد نیاز های مشتری را بهتر درک کند و با functionality سیستمی که میخوهد پیاده سازی کند بیشتر آشنا شود.



مزایای مدل نمونه ساز :

امکان تغییر و جمع آوری نیاز ها وجود دارد.

در طول تولید نرم افزار ارتباط مشتری با طراح و تولید کننده نرم افزار همواره برقرار است .

معایب مدل نمونه ساز :

- ممکن است طراح نرم افزار برای دستیابی سریع به مدل نمونه به مسائل مهمی توجه نکند مثلا سیستم عامل یا زبان برنامه نویسی.
- مشتریان چیزی را می بینند که ظاهرا یک نسخه کاری از نرم افزار است ولی نمی دانند که این نمونه اولیه با موم سره بندی شده است. نمی دانند که به لحاظ شتابی که در به کارگیری داشته ایم، کیفیت کلی نرم افزار و قابلیت نگهداری درازمدت مدنظر نبوده است.
- مهندسان نرم افزار، غالبا برای بکارگیری هر چه سریع تر نمونه اولیه، در پیاده سازی دقیق آن کوتاه می آیند.

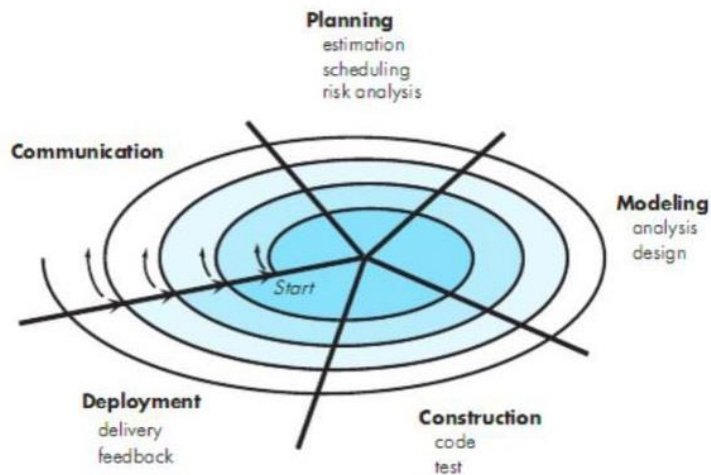
کاربرد مدل :

- هنگامی که مشتری شما خواسته ای مشروع دارد، ولی جزئیات چندانی در اختیار شما قرار نداده است.
- وقتی سازنده از بازدهی یک الگوریتم، قابلیت تطابق با یک سیستم عامل خاص مطمئن نباشد.
- گرچه از تهیه نمونه اولیه می توان به عنوان یک مدل فرایند مستقل استفاده کرد، از آن بیشتر به عنوان تکنیکی استفاده می شود که می توان در حیطه هر کدام از مدل های فرایند ذکر شده پیاده کرد.

مدل spiral

در این مدل پروژه به زیر پروژه های ریسک گرا تقسیم می شود. به این گونه که اول مقدار ریسک یا احتمال ایجاد خطا در هر زیر پروژه سنجیده می شود، سپس تیم مربوط به پیاده سازی پروژه، اول زیرپروژه هایی رو شروع می کند که احتمال ریسک آن ها بیشتر است و سپس زیرپروژه هایی را که احتمال ریسک در آن ها در درجه ی دوم قرار دارد و ... تا زمانی که تمام زیر پروژه ها انجام شوند. ریسک هایی که در این متدولوژی مد نظر هستند شامل مواردی مثل: درک نادرست از محیط مربوط به پیاده سازی پروژه، اشکالات احتمالی در کارایی برنامه، معماری ضعیف بعضی از قسمت ها یا مواردی از این قبیل.

بعد از تعیین ریسک تمام زیر پروژه هایی که برنامه ی کلی را تشکیل می دهند، پروژه توسط مدل آبشاری روند خودش را طی می کند. از این مدل زمانی استفاده می شود که معمولا هزینه و ریسک در طول پروژه مهم باشد و یا زمانی که کارفرما نسبت به نیاز های خود کاملا اطمینان نداشته باشد.



قسمتهای عمده ای که در مدل Spiral وجود دارن و باعث میشوند که ریسک اجرای یک پروژه کمتر بشود شامل موارد زیر می شوند:

1) تعریف اهداف، روشهای دستیابی به آنها و محدودیتهای موجود: در مرحله ی اول، اهداف آن مرحله، روشهای موجود برای دستیابی به آن اهداف و نیز محدودیتهای موجود به دقت بررسی شده و تشریح میشوند.

2) ارزیابی روشهای مختلف موجود: در مرحله ی دوم، روشهای مختلفی که برای دستیابی به هدف تعریف شده، مشخص شده بود بررسی میشه. معمولا در این مرحله سولوشن های مختلف مربوط به پیاده سازی این قسمت از پروژه به دقت بررسی میشه و بهترین اون انتخاب میشه و یا در بدترین حالت مشخص میشه که اون قسمت از پروژه قابل اجرا و پیاده سازی نیست.

3) تعیین و کاهش خطاهای احتمالی در سولوشن انتخاب شده: بعد از اینکه یک سولوشن برای انجام این قسمت از پروژه انتخاب شد، ریسکها و خطاهای احتمالی در این سولوشن و همچنین روشهای کاهش اونها بررسی میشه. البته در بعضی شرایط ممکن است تعیین ریسک کار بسیار مشکلی باشه.

ریسکهایی که در این قسمت مد نظر قرار می گیرن، شامل خطاهای مربوط به تکنولوژی مورد استفاده، خطاهای مربوط به پردازش این قسمت، تغییر نیازمندی ها، از دست دادن افراد کلیدی در پروژه و مسایل ایمنی باشه. (برای مثال در یک برنامه ی مربوط به موارد پزشکی ممکن است یک تصویر به غلط نشان داده شده و بیماری نادرست تشخیص داده شود.)

4) بعد از طی مراحل قبل، این قسمت از پروژه پیاده سازی میشه و یک نسخه قابل اجرا از نتایج منتشر میشه. در حقیقت این مرحله شامل پیاده سازی و تست اهدافی است که در ابتدای مرحله مشخص شد.

5) طرح ریزی مرحله ی بعد: بعد از اینکه صحت نتایج تولید شده در مرحله ی قبل تایید شد، باید مرحله ی بعد رو طرح ریزی کرد (البته اگر مرحله ای وجود داشته باشه). به علاوه باید یه مستنداتی رو در رابطه با روشهای بهبود این قسمت از پروژه تهیه کرد و همچنین باگهای مشخص شده تا کنون رو معرفی کرد تا در این قسمت و یا قسمتهای بعدی رفع بشوند.

البته در عمل معمولا از این مدل به همراه مدلهای دیگه ی توسعه ی نرم افزار استفاده میشه تا علاوه بر استفاده از مزایای مدلهای دیگه، با استفاده از این مدل ریسکهای احتمالی پروژه از همون ابتدا تشخیص داده شده و تا حد امکان کاهش داده بشوند.

مزایای مدل spiral :

- آنالیزهای زیادی روی ریسک انجام میگردد، از این رو بیشتر جلو ریسک ها را گرفته میشود.
- برای پروژههای بزرگ که مهم و حیاتی هستند، کاربردی واقع میشود.
- روی تأیید ها و مستند سازی به طور محکمی کنترل انجام میشود.
- Functionality های بیشتر میتواند بعدا به پروژه اضافه شود.
- نرم افزار در چرخه های اول مدل نرم افزاری تهیه میشود.

معایب روش spiral :

- میتونه یک مدل هزینه بر باشد.
- برای آنالیز کردن ریسک ها به تخصص خاصی نیاز هست.
- موفقیت پروژه تا حد زیادی به فاز آنالیزهای مربوط به ریسک وابسته میشود.
- برای پروژه های کوچک کاربردی نیست.

کاربرد روش spiral :

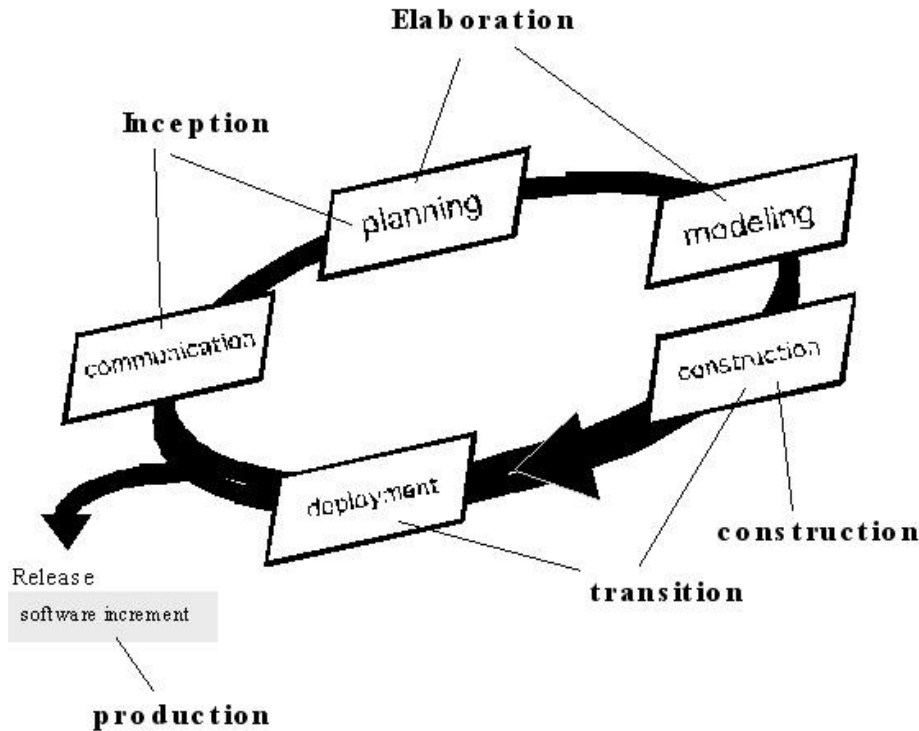
- وقتی هزینه و ارزیابی risk ها مهم هست.
- برای پروژهایی که ریسک در آن ها در حد متوسط یا زیاد هست.
- کاربران در مورد نیاز هایشان به کاملاً مطمئن نیستند.
- نیازمندی ها پیچیده هست.
- تغییرات قابل توجهی مورد انتظار هست. (تحقیقات و اکتشافات)

مدل Unified Process (RUP)

این فرآیند یک روش نظام مند برای تخصیص کارها و مسئولیت‌ها در یک تیم توسعه نرم‌افزار ارائه می‌دهد و هدف آن تولید نرم‌افزار بصورت بهینه و با کیفیت بالاست که بتواند نیازهای کارفرما را تحت یک برنامه زمانی مشخص و با بودجه قابل پیش‌بینی برآورده سازد. آر. یو. پی بهره‌وری تیم تولید نرم‌افزار را با فراهم نمودن دسترسی تمام افراد تیم به یک پایگاه دانش سهل‌الوصول به همراه راهنماها، الگوها و ابزارهای کمکی برای همه فعالیت‌های حیاتی توسعه، افزایش می‌دهد. از آنجا که تمام افراد به منابع یکسانی دسترسی دارند، لذا دید مشترکی برای توسعه نرم‌افزار برخوردار هستند.

آریوپی امکان استفاده موثرتری از زبان یکپارچه مدلسازی (UML) را فراهم می‌سازد (دقت شود که در عین حال آر. یو. پی و یو. ام. ال کاملاً مستقل از یکدیگر هستند و نباید آنها را با هم یکی فرض کنیم). به کمک تکنیک‌های آر. یو. پی بخش‌های عمده‌ای از فرآیند تولید نرم‌افزار به طور خودکار انجام شده و همچنین استفاده از مدل‌های تولید شده در فرآیندهای گذشته در پروژه‌های جاری به سادگی امکان‌پذیر است. این فرآیند با موقعیت‌های مختلف تطبیق یافته و برای سازمانهای بزرگ یا حتی کوچک تولید و توسعه نرم‌افزار قابل استفاده است.

RUP کلیه مراحل انجام یک پروژه شامل تحلیل سیستم، برنامه‌ریزی، بررسی ریسکها، تولید و تست نرم‌افزار را در بر می‌گیرد و چهارچوبی در جهت انجام صحیح و موفق پروژه‌های نرم‌افزاری فراهم می‌سازد.



مزایا:

1. تسهیل توسعه تکراری نرم افزار
2. مدیریت نیازها
3. مدل کردن تصویری نرم افزار
4. بازبینی کیفیت نرم افزار
5. کنترل تغییرات در نرم افزار
6. امکان استفاده از طریق وب

معایب:

1. اعضای تیم در این مدل باید در کار خود حرفه ای باشند.
2. روند این پروسه خیلی پیچیده و بدون نظم میباشد.
3. در پروژه های مهمی که از تکنولوژی های جدید استفاده میکنند، استفاده مجدد از components ها غیر ممکن میشود.
4. یکپارچگی در روند ساخت یک نرم افزار به نظر چیز آسانی هست. ولی در بخصوص پروژه های بزرگ با چندین جریان development، این مدل فقط پیچیدگی ایجاد میکند و مشکلات بیشتری در مرحله test به وجود می آورد.

مدل evolutionary :

مثل هر سیستم پیچیده ای، برنامه های نرم افزار هم با گذشت زمان نیاز به تکامل پیدا میکنند. به علت اینکه نیازها تغییر میکنند، این که یک مسیر خطی بخواهیم داشته باشیم برای روند این کار غیر ممکن هست. Deadline هایی که برای اتمام پروژه ها گذاشته میشود از اینکه به طور کامل و جامع بخواهیم یک نرم افزار داشته باشیم جلوگیری میکنند. در این موارد برای اینکه یک سیستمی را داشته باشیم که بتواند رشد کند و تغییر کند، از مدل های evolutionary استفاده میشود که به صورت تکرار در حلقه هایی هستند. از انواع روش های evolutionary میتوان به prototyping و spiral اشاره کرد.

مدل concurrent :

مدل فرآیند همزمان یک سری رویدادهایی را مشخص می سازد که انتقال از حالتی به حالت دیگر را برای هر کار مهندسی نرم افزار آغاز می کنند.

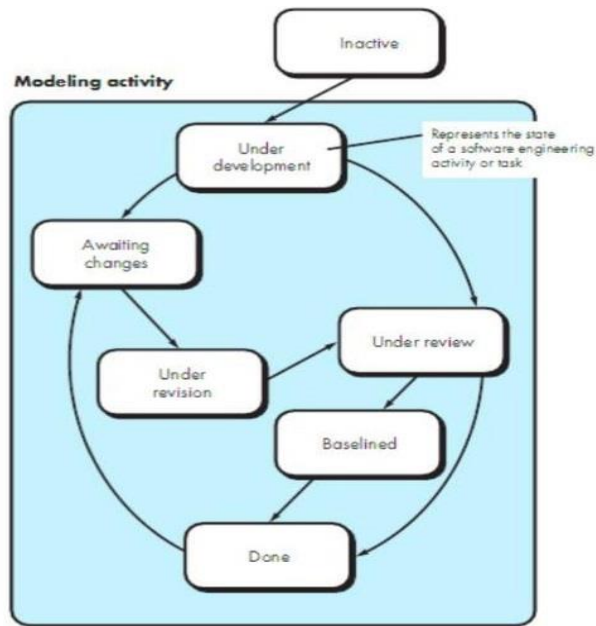
مدل فرآیند همزمان اغلب بعنوان معیاری برای توسعه و تولید برنامه های کاربردی خادم/مخدوم استفاده می شود. یک سیستم خادم/مخدوم متشکل از مجموعه ای اجزای کاربردی است. وقتی مدل فرآیند همزمان در خادم/مخدوم استفاده می شود، فعالیت هایی را در دو بعد تعریف می کند، بعد سیستم و بعد مؤلفه. موضوعات مربوط به سیستم با استفاده از سه فعالیت مورد توجه قرار می گیرند، طراحی، مونتاژ و کاربرد. بعد جزء از نظر دو فعالیت مورد بررسی است: طراحی و شناسایی و همزمانی به دو شکل حاصل می شود:

1- فعالیت های سیستم و جزء که بطور همزمان رخ می دهند و می توان آنها را با استفاده از رهیافت وضعیت گرا مدل سازی کرد.

2- یک برنامه کاربردی معمول خادم/مخدوم بوسیله اجزا بسیاری به اجرا در می آید که هر کدام از آنها را می توان بطور همزمان طراحی و شناسایی نمود. مدل فرآیند همزمان در مورد همه نوع تولید نرم افزار قابل اجراست و تصویر دقیقی از وضعیت کنونی پروژه را ارائه می دهند.

ساختار مدل

- شکل زیر طراحی از یک فعالیت با مدل توسعه همروند ارائه می دهد. این فعالیت- مدل سازی- ممکن است در هر زمان مفروض، در یکی از حالت های ذکر شده باشد. (حالت، رفتاری از سیستم است که از بیرون قابل مشاهده است)
- به طور مشابه، فعالیت های دیگر مانند ارتباطات و ساخت را می توان به شیوه ای مشابه نمایش داد. همه فعالیت ها به صورت همروند وجود دارند ولی در حالت های متفاوت قرار می گیرند.
- 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 شوند.



در حالت موازی، زمان کاهش می یابد ولی ممکن است بعضی پارامترهای دیگر مثل دوباره کاری و مدیریت پروژه و ... افزایش یابد.

مزایا:

- تصویر دقیقی از موقعیت فعلی و پیشرفت پروژه میدهد.
- این مدل برای همه ی گونه های software development قابل استفاده است.

معایب:

- این مدل نیازمند وجود ارتباط تنگاتنگ و زیادی بین همه ی تیم ها است که ممکن است گاهی قابل تحقق نباشد.

مدل formal :

- مجموعه ای از فعالیت ها که به مشخص کردن ریاضی و رسمی نرم افزار کامپیوتری منجر می شود.
- مدل روش های رسمی گرچه عمومیت ندارد اما نوید نرم افزاری عاری از نقص را می دهد.
- توسعه ی این مدل ها در حال حاضر وقت گیر است و به آموزش گسترده نیاز مند است.
- ارتباط با مشتری که دید فنی ندارد دشوار است.
- مناسب برای ساخت نرم افزارهای ایمنی حیاتی مثل نرم افزارهای دستگاه های پزشکی و هوافضا یا نرم افزارهایی که در صورت بروز خطا در آن ها دستخوش زیان های اقتصادی کلان می شوند.

مزایا :

وضوح ، روشنی ، سازگاری ، جامع بودن

معایب :

مصرف زمان بیشتر ، هزینه ی بیشتر ، آموزش های سخت تر ، عدم مکانیزم مناسب در برقراری ارتباط با مشتری ، استفاده ی آن در سیستم هایی با حساسیت بالا.

فرایند نرم افزار شخصی (Personal Software Process)

در مدل فرایند های نرم افزار شخصی پنج فعالیت چارچوبی تعریف می شود:

مراحل :

- برنامه ریزی
 - خواسته ها شناسایی می شود و برآورد منابع و تعیین اندازه پروژه انجام می شود.
 - وظایف لازم برای توسعه نرم افزار تعیید و زمان بندی پروژه انجام می شود.
- طراحی سطح بالا
 - مشخصات خارجی برای هر کدام از مولفه هایی که قرار است تعیین شود و طراحی مولفهانجام می شود.
 - نمونه های اولیه ساخته می شوند، همه مسائل و مشکلات ثبت و پیگیری می شوند.
- مرور طراحی سطح بالا
 - روش های وارسی رسمی برای یافتن خطاهای طراحی اعمال می شوند.
 - معیارهای مربوط به وظایف مهم و نتایج کار نگهداری می شوند.
- توسعه
 - طراحی سطح بالا پالایش و بازبینی می شود .کدها تهیه، بازبینی، کامپایل و آزموده می شوند.
- پایان کار
 - با استفاده از معیارها وموازن جمع آوری شده، اثر بخشی فرایند تعیین می شود.

مزایا:

- با استفاده از این مدل میتوان در بازه ی زمانی کوتاهتری، کار بیشتری را با کمترین error انجام داد.
- استفاده از این مدل منجر به زمان بندی های دقیقی برای پروژه میشود.

معایب:

- یادگیری استفاده از این مدل نسبتا طو نی و پرهزینه است.
- بیان measurement های خواسته شده در این مدل، برای بسیاری از مهندسين نرم افزار دشوار است.

فرایند نرم افزار تیمی (Team Software Process)

- برای انجام پروژه های نرم افزاری در پایه ی صنعتی را تیمی از دست اندرکاران انجام می دهند. هدف فرایند نرم افزار تیمی تشکیل تیم پروژه ی خود هدایت گر است. فعالیت های زیر را برای فرایند نرم افزار تیمی تعریف می کنند.
- تشکیل تیم خود هدایت گری که کار خود را برنامه ریزی و پیگیری می کند، اهداف را تعیین می کند و خود به تعیین فرایند و طرح ها اقدام می کنند.
 - نشان دادن شیوه ی راهبردی و ایجاد انگیزه در تیم ها به مدیران و چگونگی کمک به آنها در حفظ حداکثر کارایی.
 - شتاب بخشیدن به بهبود فرایند نرم افزاری.
 - فراهم ساختن دستور بهسازی برای سازمان های بالغ.
 - تسهیل آموزش دانشگاهی مهارت های تیمی در سطح صنعتی.

مزایا:

- تقریباً مانند مدل قبلی در برنامه ریزی های دقیق به اعضای تیم کمک میکند.
- استفاده از مدل منجر به تولید نرم افزار هایی با کیفیت با میشود.

معایب:

- از آنجا که استفاده از این مدل نیز نیاز به شناخت و یادگیری آن باری همه ی اعضای تیم دارد و این فرآیند زمان گیر و پرهزینه خواهد بود.

مدل انتخابی:

ما در این پروژه مدل prototyping را انتخاب کردیم. در این مدل به علت اینکه نمونه ای از سیستم تهیه شده، کاربر پیش زمینه بهتری نسبت به سیستم ای که به او قرار هست تحویل داده شود، پیدا می کند. چون افراد زیادی قرار هست که از این سیستم استفاده کنند، این که این سیستم به شکلی باشد که همه با آن ارتباط برقرار کنند، مهم می باشد. و با طراحی نمونه میتوانیم به درخواست کاربر نزدیک تر شویم. و همچنین interface مطابق با سلیقه کاربر تهیه کنیم.

دلیل انتخاب مدل

در پروژه Degree Overview به علت این که web base هست و یک سیستم online نیاز مند این هست که کاربران به راحتی بتوانند با سیستم ارتباط برقرار کرده و نیاز های خود را برطرف کنند، از این رو در این مدل تلاش می شود که سیستم به سطحی برسد که کاربر با کمترین زمان نیاز های خود را برطرف کند.

از مزایای این مدل که با پروژه ما مطابقت دارد:

- مشکلات سریع تر پیدا می شوند.

- با بازخورد های مشتری جواب های بهتری به مشکلاتمان پیدا میکنیم.
- اگر function ای فراموش شده که توسط کاربر معرفی شود، با این نمونه ها کمبود ها برطرف خواهد شد.
- Function هایی که پیچیده و مشکل هستند، مشخص میشوند. که بررسی های بیشتری بر روی آن ها انجام شود.

توضیح فاز های مدل بر روی پروژه

فاز های مدل **prototyping** عبارت اند از :

Communication 🚩

در این فاز تیم با ذی نفعان پروژه جلسه ای باید داشته باشیم که دید کلی راجع به اهدافی که سیستم میخواهد ارائه دهد، صحبت شود، به شکلی که ما بتوانیم نیاز مندی های سیستم را متوجه شویم. در این پروژه اهداف و توضیح هات این پروژه به صورت متنی به گروه ما واگذار شد.

ما با بررسی متن ارائه شده در <http://score-contest.org/2016/projects/degreeov.php> نیاز مندی های سیستم را در قسمت 4 مستندات ثبت کردیم.

Quick Design 🚩

در این فاز یک design سریع باید تولید شود که براین مبنا هست که یک نمایش از جنبه های دیداری سیستم تهیه کنیم و صرفاً فقط ظاهر سیستم را به نمایش می گذارد. همان user interface که قرار هست end user مشاهده کند.

Construction of prototyping 🚩

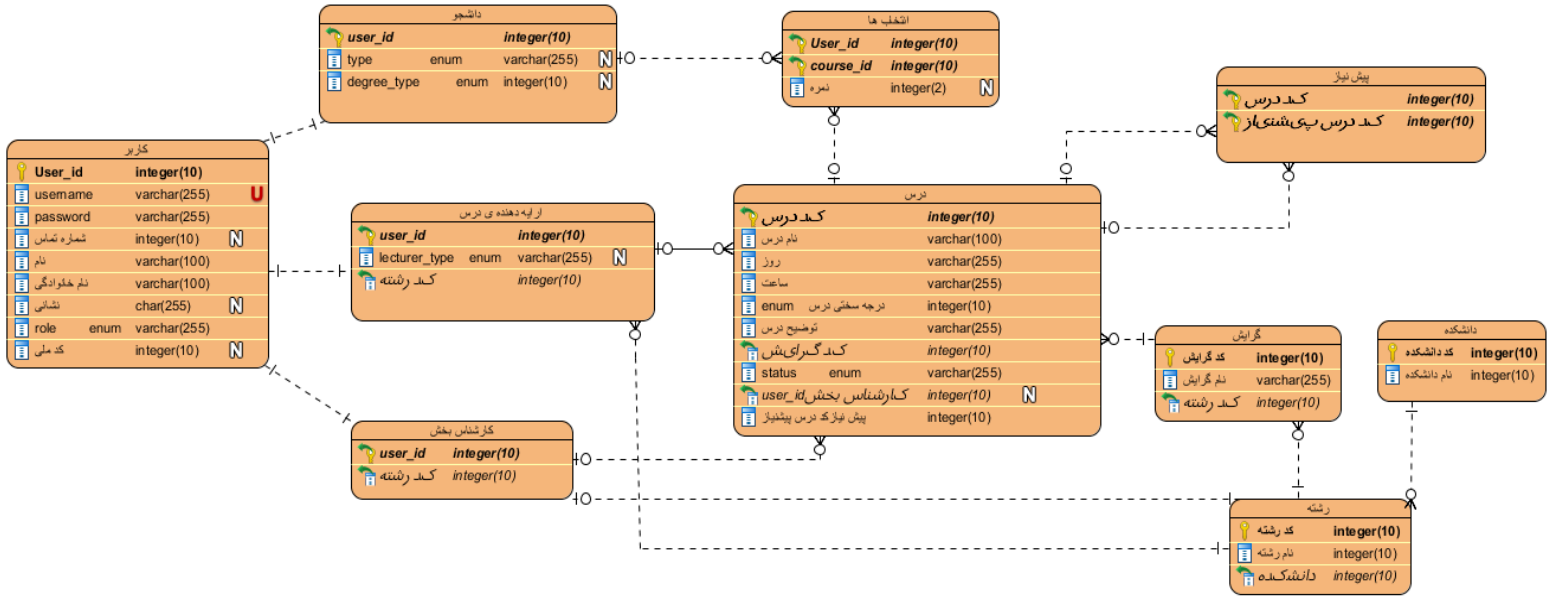
در این فاز ما به این شکل عمل میکنیم که با توجه به ظواهری که در فاز قبلی شروع کردیم، در این فاز در iteration های اولیه بودن توجه به جزئیات کامل سعی می کنیم کلیاتی را به صورت سریع داشته باشیم و به صورت آهسته شروع میکنیم به پیاده سازی تا به سیستم نهایی برسیم. این کار در iteration هست به شکلی که این سیستم بازخوردی از مشتری به ما بدهد که بتوانیم اگر تغییراتی را نیاز داشت که stakeholder درخواست دارد را پیاده سازی کنیم. و این کار به آخر نیافتد که stakeholder دیدی از سیستم نداشته باشد.

Deployment Delivery & feedback 🚩

در این فاز **prototype** ای که در مرحله قبل طراحی شد به را **deploy** می کنیم که توسط **stakeholder** ارزیابی شود و با بازخورد او سیستم را به نحوی که مورد نظر اوست بهبود داده و تغییرات لازم را انجام دهیم و همچنین به ما این امکان را می دهد که درک بهتری از کار هایی که باید روی سیستم انجام دهیم، داشته باشیم.

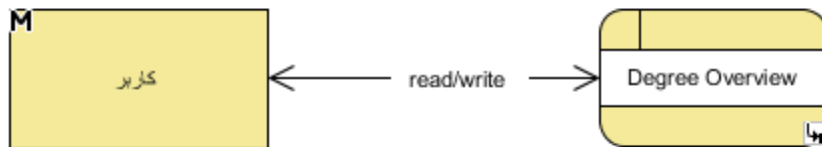
این اعمال به این شکل در **iteration** های متوالی قرار میگیرند تا به سیستم نهایی برسیم و به **stakeholder** تحویل دهیم.

ERD

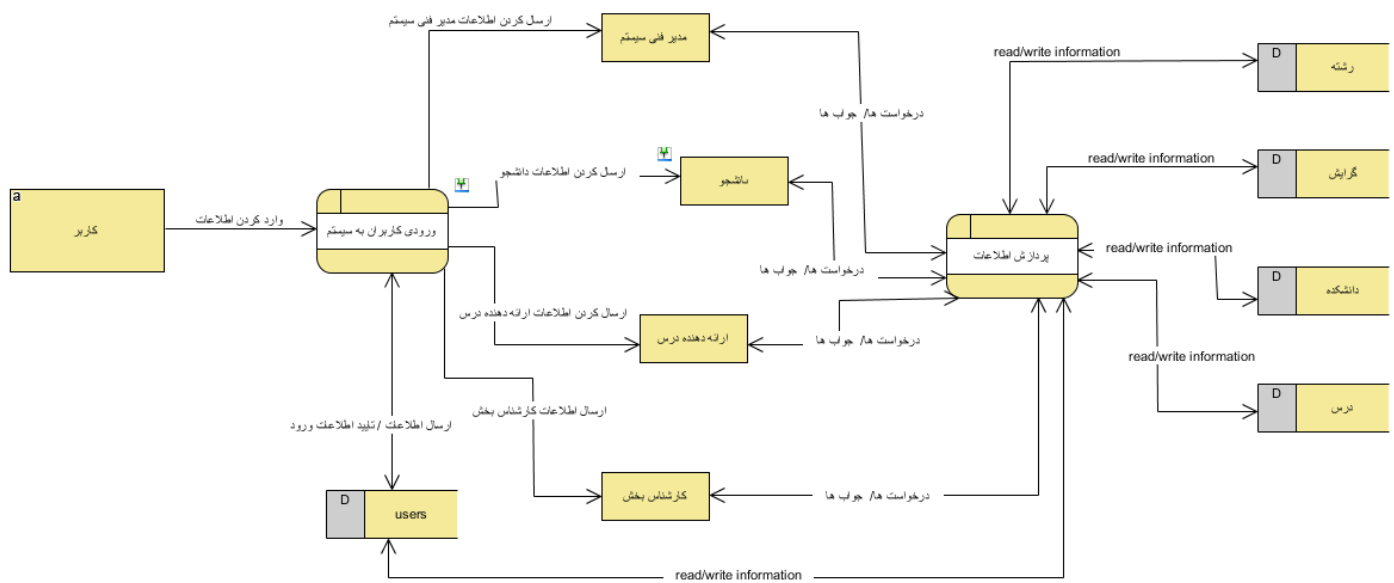


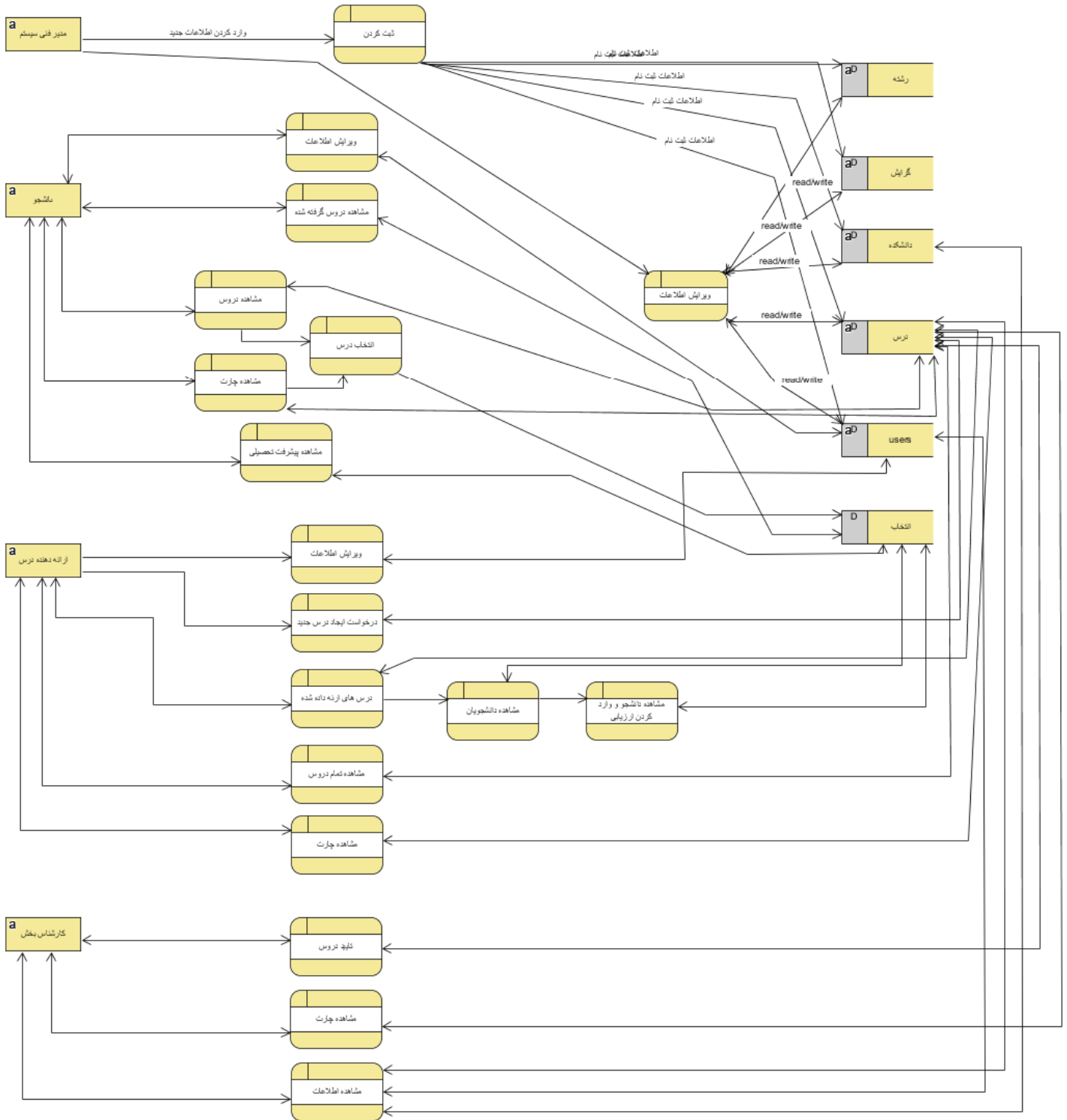
DFD

سطح 0

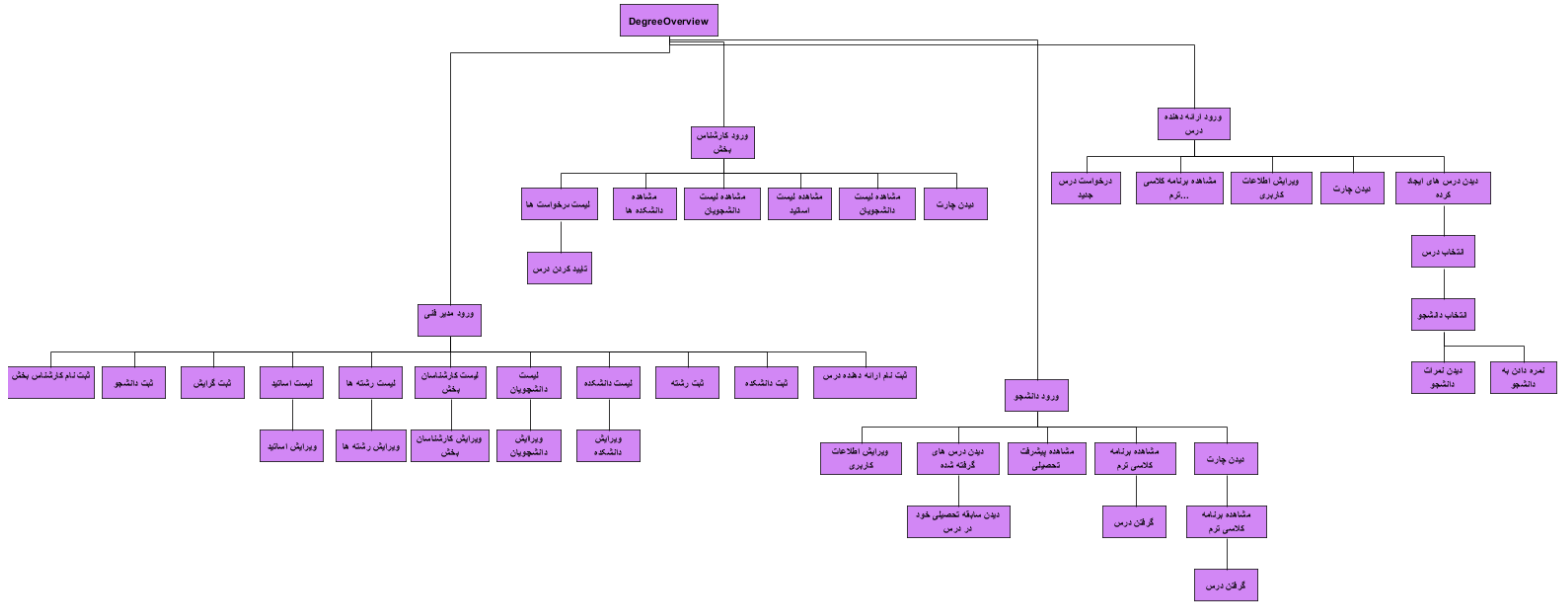


سطح 1





Function Hierarchy Diagram (FHD)



اعضای گروه:

کسانی که روی این پروژه کار میکنند، فاطمه رحیمی و مریم صباحی می باشند.

تا اینجا تمام کارها با همکاری و مشورت با یکدیگر انجام شده.

Recourses:

Software Engineering: A Practitioner's Approach, Pressman

<https://en.wikipedia.org>

<http://istqbexamcertification.com/>

[https://msdn.microsoft.com/en-us/library/aa771426\(v=bts.10\).aspx](https://msdn.microsoft.com/en-us/library/aa771426(v=bts.10).aspx)

<http://mjahangir.ir/MyFile/UniCourse/SE1922/SE1-Session4-92.11.29.pdf>

<http://www.my-project-management-expert.com/the-advantages-and-disadvantages-of-rup-software-development.html>