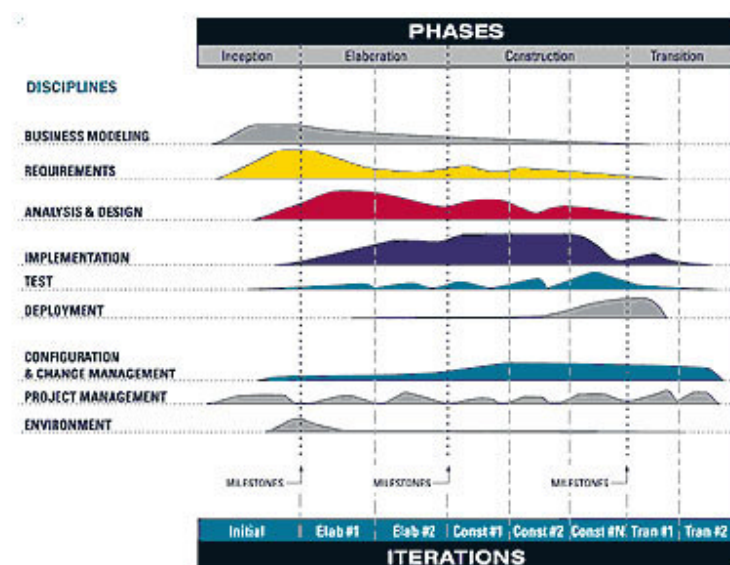


شرکت البرز تدبیر سامانه (سهامی خاص)
خدمات انفورماتیک (مشاوره، طراحی و پیاده سازی سیستم ها)

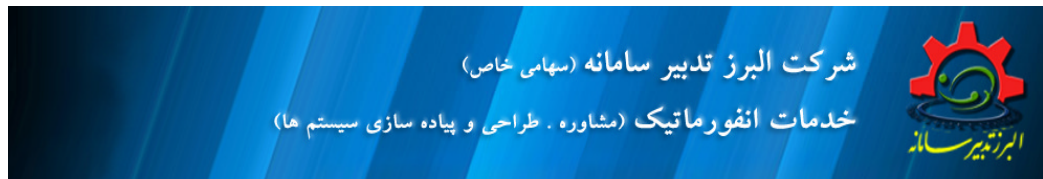
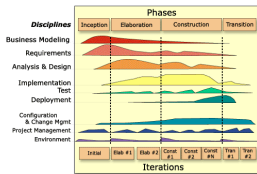
RUP چیست؟



معماری و ساختار کلی RUP

فرایند انجام یک پروژه تعریف می‌کند که چه کسی، چه کاری را در چه هنگام و چگونه برای رسیدن به هدف (انجام پروژه) انجام می‌دهد. در مهندسی نرم‌افزار، هدف ساختن یک محصول نرم‌افزاری و یا بهبود یک نمونه‌ی موجود است. هدف از تعیین فرایند، تضمین کیفیت نرم‌افزار، برآورده شدن نیازهای کاربر و قابل تخمین بودن زمان و هزینه‌ی تولید می‌باشد.

علاوه بر این، تعیین فرایند، روندی جهت تحویل مصنوعات دوران تولید نرم‌افزار به کارفرما و ناظر پروژه ارائه می‌دهد تا از این طریق اطمینان حاصل کنند که پروژه روند منطقی خود را طی می‌کند و نظارت درست بر



انجام پروژه ممکن است و از سوی دیگر، معیاری برای ارزیابی پروژه انجام شده می‌باشد. تا کنون متدولوژی‌های مختلفی برای فرآیند تولید نرم‌افزار ارائه شده‌اند که یکی از مشهورترین آنها RUP است.

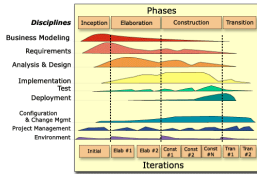
RUP، متدولوژی ارائه شده توسط شرکت Rational، پرکاربردترین فرآیند تولید و توسعه نرم‌افزاری در دنیای کنونی است و به عنوان یک استاندارد صنعتی بالفعل در دنیای IT پذیرفته شده است. به گزارش رویتر در سال ۲۰۰۱ میلادی بیش از ششصد هزار شرکت تولید کننده نرم‌افزار، از ابزارهای شرکت Rational استفاده می‌کرده‌اند که این تعداد کماکان هم در حال افزایش است.

این متدولوژی، برای انواع پروژه‌های نرم‌افزاری در دامنه‌های مختلف (مانند سیستم‌های اطلاعاتی، سیستم‌های صنعتی، سیستم‌های بلادرنج، سیستم‌های تعبیه شده، ارتباطات راه دور، سیستم‌های نظامی و ...) و در اندازه‌های متفاوت، از پروژه‌های بسیار کوچک (یک نفر در یک هفته) تا پروژه‌های بسیار بزرگ (چند صد نفر تولید کننده با پراکندگی جغرافیایی)، کاربرد دارد.

مزیت بزرگ این متدولوژی، استفاده از روش تکرار در تولید و مدیریت تولید نرم‌افزار است که این امر، امکان تولید مبتنی بر کاهش ریسک و مواجهه با مشکلات اصلی در ابتدای کار و در نتیجه احتمال موفقیت بیشتر را فراهم می‌کند. از محاسن دیگر این متدولوژی مبنا قرار دادن نرم‌افزار و تولید یک معماری پایدار در ابتدای کار است، که در نتیجه امکان کشف مشکلات عمده ساختاری، تست و مجتمع سازی ممتد را از ابتدای کار فراهم می‌کند. از دیگر مزایای این روش این است که افراد تیم همزمان با پیشرفت پروژه، مطالب جدیدی فرا می‌گیرند و کیفیت فرآیند تولید نیز به طور مرتب افزایش می‌یابد.

RUP دارای دو بعد است:

- ۱- محور افقی نشان دهنده‌ی زمان است و با پیشرفت خود جنبه‌های چرخه‌ی حیات فرآیند و فازهای RUP را نشان می‌دهد.
- ۲- محور عمودی نمایانگر دیسپلین‌های RUP است که فعالیت‌ها را با استفاده از ماهیتشان به صورت منطقی دسته‌بندی می‌کند.



در هر فاز ممکن است یک یا چند تکرار وجود داشته باشد و در هر تکرار عملیات دیسپلین‌های مختلف انجام می‌گیرند.

فازها و milestone های یک پروژه در RUP

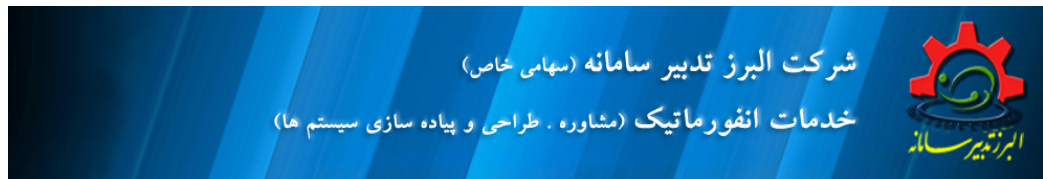
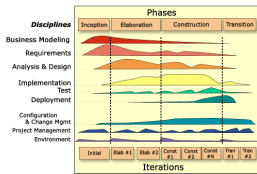
۱- آغازین (Inception)

هدف اصلی این فاز دستیابی به توافق میان کلیه ذینفعان بر روی اهداف چرخه‌ی حیات پروژه است. فاز Inception به دلیل تلاشهای تولید و توسعه جدید به صورت پایه‌ای اهمیت فراوانی دارد که در آن ریسک‌های نیازسنجی و تجاری مهمی وجود دارد که باید پیش از اینکه اجرای پروژه مورد توجه قرار گیرد، بررسی شوند. برای پروژه‌هایی که بر توسعه سیستم موجود متمرکزند، فاز Inception کوتاهتر است، با اینحال این فاز برای حصول اطمینان از اینکه پروژه ارزش انجام دادن دارد و امکان‌پذیر نیز هست، انجام می‌شود. اهداف اصلی فاز آغازین شامل موارد زیر است:

- بدست آوردن محدوده نرم‌افزاری پروژه و محدودیت‌های آن که شامل یک دید عملیاتی، معیار پذیرش و اینکه چه چیز باید در محصول باشد و چه چیز نباید باشد، می‌شود.
- مشخص کردن Use-Case های اساسی سیستم، سناریوهای اصلی عملیات که مسائل مربوط به طراحی اصلی را ایجاد می‌کند.
- نمایش و شاید توضیح حداقل یک معماری کاندیدا برای بعضی سناریوهای اصلی
- برآورد هزینه و زمان کلی برای کل پروژه

۲- (جزئیات) Elaboration

هدف فاز جزئیات تعیین معماری کلی سیستم به منظور فراهم آوردن یک زمینه‌ی مناسب برای قسمت عمده‌ی طراحی و پیاده‌سازی در فاز Construction است. معماری با در نظر گرفتن بیشتر نیازمندی‌های مهم (آن دسته از نیازمندی‌ها که تأثیر زیادی بر معمار سیستم دارد) و نیز ارزیابی ریسک کامل می‌شود.



پایداری معماری از طریق یک یا چند نمونه‌ی اولیه ساختاری ارزیابی می‌شود. اهداف اصلی فاز جزئیات شامل موارد زیر است:

- اطمینان از اینکه معماری، نیازمندی‌ها و طرح‌ها به اندازه‌ی کافی پایداری و ریسک‌ها به اندازه‌ی کافی کاهش یافته‌اند بطوریکه بتوان هزینه و زمان‌بندی لازم برای تکمیل تولید را پیش‌بینی کرد. برای اکثر پروژه‌ها، گذر از این مرحله‌ی مهم مانند انتقال از یک عملیات سبک و سریع و با ریسک پایین به یک عملیات با هزینه و ریسک بالا همراه با اجبار سازمانی است.
- بیان همه‌ی ریسک‌های پروژه که از نظر ساختاری اهمیت دارند.
- ایجاد یک معماری پایه، مشتق شده از سناریوهای مهم که از لحاظ ساختاری اهمیت دارند، که این معماری ریسک‌های فنی عمده پروژه را نیز مشخص می‌کند.
- تولید یک نمونه‌ی اولیه‌ی تکاملی از مولفه‌های با کیفیت تولیدی خوب، و همچنین یک یا چند نمونه‌ی اولیه‌ی اکتشافی و نمونه‌های اولیه‌ی غیر قابل استفاده جهت کاهش ریسک‌های خاص مانند:

* سازش‌های مربوط به نیازمندی‌ها یا طراحی

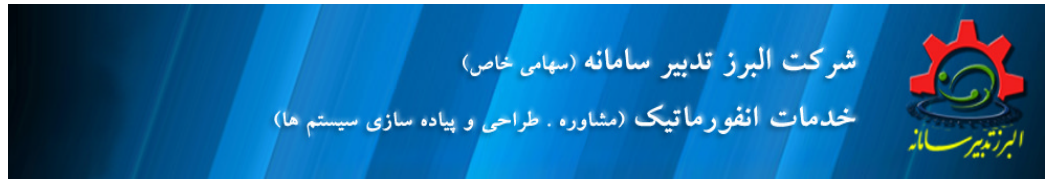
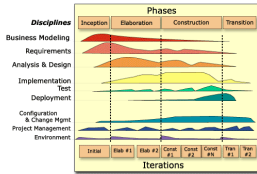
* استفاده‌ی مجدد از مؤلفه‌ها

* عملی بودن محصول یا توضیحات برای سرمایه‌گذاران، مشتریان و کاربران نهایی

- توضیح اینکه معماری پایه از نیازمندی‌های سیستم با هزینه‌ی منطقی و در زمان منطقی پشتیبانی می‌کند
- ایجاد یک محیط پشتیبانی کننده

۳- ساخت (Construction)

هدف این فاز واضح سازی نیازمندی‌های باقیمانده و تکمیل تولید سیستم بر اساس معماری مبنا می‌باشد. فاز ساخت به نوعی یک فرآیند ساخت است که در آن تأکید بر مدیریت منابع و کنترل عملیات به منظور بهینه‌سازی هزینه‌ها، زمان‌بندی‌ها و کیفیت است. در این حالت یک انتقال از تولید یک نمونه‌ی ذهنی در



طی فازهای Inception و Elaboration به تولید محصولات قابل استقرار در طی Construction و Transition می‌شود. اهداف اصلی فاز Construction شامل موارد زیر می‌باشد:

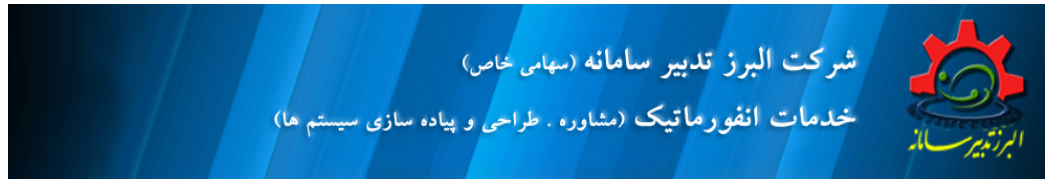
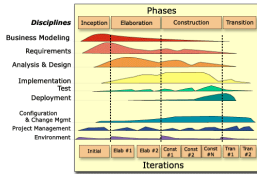
- کمینه کردن هزینه‌های تولید با بهینه‌سازی منابع و پرهیز از دور انداختن و دوباره‌کاری غیر ضروری
- دستیابی هرچه سریعتر به کیفیت کافی
- دستیابی هر چه سریعتر به ویرایش‌های مفید (آلفا، بتا و سایر نسخه‌های تست)
- کامل کردن تحلیل، طراحی، تولید و تست کارآیی مورد نیاز
- تولید تکراری و گام به گام به محصول کامل که آماده‌ی انتقال به محیط کاربران باشد
- تصمیم در مورد اینکه آیا نرم‌افزار، سایت‌ها و کاربران همه برای استقرار طرح آمادگی دارند
- دستیابی به میزانی از موازی سازی در کار تیم‌های تولید

۴- انتقال (Transition)

تمرکز این فاز بر این است که تضمین نماید نرم‌افزار برای کاربران نهایی آماده می‌باشد. فاز Transition می‌تواند به چندین تکرار تقسیم شود، و شامل تست کردن محصول برای آماده‌سازی جهت انتشار و ایجاد تنظیمات کوچک بر اساس بازخورد کاربر می‌باشد. در این نقطه از چرخه‌ی حیات، بازخورد کاربر باید بطور عمده بر تنظیم دقیق محصل، پیکربندی، نصب و نکات مربوط به قابلیت استفاده تمرکز یابد، و همه‌ی نکات ساختاری اصلی باید هرچه زودتر در چرخه‌ی حیات پروژه طرح شوند.

با به اتمام رسیدن فاز Transition اهداف چرخه‌ی حیات باید برآورده شده باشند و پروژه در موقعیتی باشد که بتوان آنرا خاتمه داد. در برخی موارد، پایان چرخه‌ی حیات فعلی ممکن است با آغاز چرخه‌ی حیات بعدی در مورد همان محصول همزمان شود و ما را به سمت تولید یا ویرایش دیگری هدایت کند. برای پروژه‌های دیگر، پایان فاز Transition ممکن است با تحویل کامل خروجی‌ها به گروه سومی که ممکن است مسؤول عملیات نگهداری و پیشرفت سیستم تحویل داده شده می‌باشند، همزمان شود. این فاز بر اساس نوع محصول در فاصله‌ی بسیار ساده تا بی‌نهایت پیچیده قرار دارد.

نصب یک نسخه‌ی جدید از یک بسته نرم‌افزاری موجود ممکن است بسیار ساده باشد، در حالیکه جایگزینی سیستم کنترل ترافیک هوایی یک کشور ممکن است بسیار پیچیده باشد. فعالیت‌هایی که در طول یک تکرار



در فاز Transition انجام می‌گیرد به هدف بستگی دارند. برای مثال معمولاً در هنگام رفع اشکالات، پیاده‌سازی و تست کافی هستند. با این وجود اگر ویژگی‌های جدیدی باید اضافه شوند، این تکرار شبیه به تکراری در فاز Construction می‌شود که نیازمند تحلیل و طراحی و غیره است.

فاز Transition زمانی وارد عمل می‌شود که یک خط مبنا آنقدر بالغ شده که بتواند در دامنه‌ی کاربر نهایی استقرار یابد. این امر بطور نمونه نیازمند این است که تعدادی زیر مجموعه‌ی قابل استفاده از سیستم با کیفیت قابل قبول و مستندات کاربر، کامل شده باشند، تا انتقال به کاربر نتایج مثبتی را برای همه‌ی گروه‌ها در بر داشته باشد. اهداف مهم فاز Transition عبارتند از:

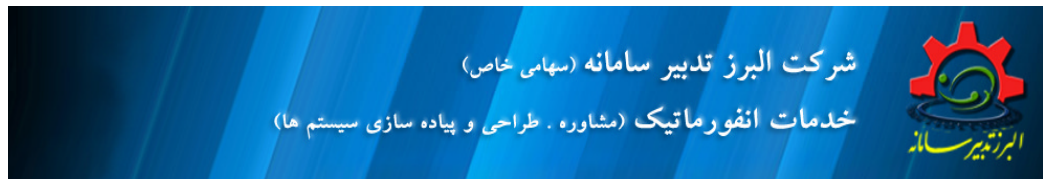
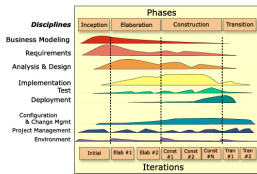
- تست بتا برای تشخیص اعتبار سیستم جدید با توجه به انتظارات کاربر
- تست بتا و عملیات موازی همراه با یک سیستم قدیمی که در حال جایگزینی می‌باشد.
- تبدیل پایگاه‌های داده‌ی عملیاتی
- آموزش کاربران و نگهداری کنندگان
- بازاریابی، توزیع و فروش برای نخستین انتشار محصول
- تنظیم فعالیت‌ها از قبیل رفع اشکال، افزایش کارایی و قابلیت استفاده
- ارزیابی خط مبناهای استقرار در مقایسه با تصویر کلی و معیار قابلیت قابل قبول برای محصول
- دستیابی به موافقت ذینفع در مورد اینکه خط مبناهای استقرار کامل می‌باشند
- دستیابی به موافقت ذینفع در مورد اینکه خط مبناهای استقرار با معیار ارزیابی تصویر کلی سازگارند

دیسپلین‌های RUP

دیسپلین مجموعه‌ای از کارهای به هم مرتبطی است که برای انجام جنبه خاصی از یک پروژه انجام می‌شوند. متدولوژی RUP دارای ۶ دیسپلین اصلی (مربوط به تولید محصول) و ۳ دیسپلین کمکی (مربوط به تیم و محیط تولید) است که در ادامه به ترتیب معرفی خواهند شد.

مدل‌سازی کسب و کار (Business Modeling)

هدف مدل‌سازی کسب و کار عبارتند از:



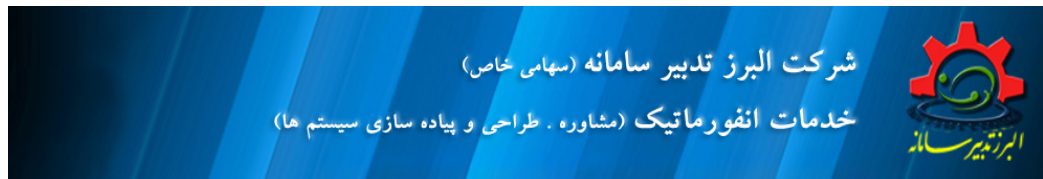
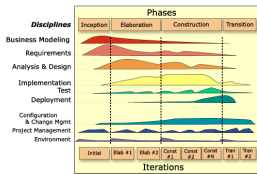
- شناخت ساختار و دینامیک‌های سازمانی که در آن یک سیستم باید استقرار یابد (سازمان هدف).
- شناخت مشکلات فعلی در سازمان هدف و تشخیص پتانسیل‌های بهبود
- تضمین اینکه مشتری، کاربر نهایی و تولید کنندگان یک شناخت مشترک از سازمان هدف دارند .
- هدایت نیازمندی‌های سیستم که برای حمایت از سازمان هدف مورد نیازند .
- دیسیپلین مدل‌سازی کسب و کار توضیح می‌دهد که برای رسیدن به این هدف چگونه می‌توان یک تصویر کلی از سازمان را تولید نمود، و براساس این تصویر کلی فرآیندها، نقش‌ها و مسؤولیت‌های آن سازمان را در یک مدل Use-case کسب و کار و یک مدل شیء کسب و کار تعریف کرد.

نیازمندی‌ها (Requirements)

اهداف دیسیپلین نیازمندی‌ها عبارتند از:

- تشخیص و نگهداری موارد توافق با مشتری‌ها و سایر ذینفعان در مورد کارهایی که سیستم باید انجام دهد .
- فراهم آوردن شناخت بهتر از نیازمندی‌های سیستم برای تولید کنندگان سیستم
- تعریف مرزهای و حدود سیستم
- فراهم کردن یک پایه برای طرح ریزی مفاهیم تکنیکی تکرارها
- فراهم کردن یک پایه برای تخمین مخارج و زمان تولید سیستم
- تعریف یک واسط کاربر برای سیستم با تمرکز بر روی نیازها و اهداف کاربران

برای دستیابی به این اهداف، ابتدا فهم تعریف و محدوده‌ی مسأله‌ای که سعی داریم با این سیستم آن را حل کنیم، حائز اهمیت می‌باشد. قوانین کسب و کار مدل Use-Case کسب و کار و مدل شیء کسب و کار که در طول مدل‌سازی کسب و کار تولید شده به عنوان ورودی با ارزشی برای این تلاش خواهند بود. در این راستا ذینفعان تشخیص داده می‌شوند و درخواستهای ذینفعان استخراج، جمع‌آوری و تجزیه و تحلیل می‌شوند. یک مستند تصویر کلی، یک مدل Use-Case، Use-Case ها و مشخصه‌های تکمیلی برای توضیح کامل سیستم تولید می‌شود. این توضیح در واقع کاری را که سیستم انجام خواهد داد بیان می‌کند.



این مستندات بعنوان منابع مهم اطلاعات تولید می‌شود. در تولید این مستندات باید خواسته‌های همه ذینفعان را در نظر گرفت.

تحلیل و طراحی (Analysis & Design)

اهداف تحلیل و طراحی عبارتند از:

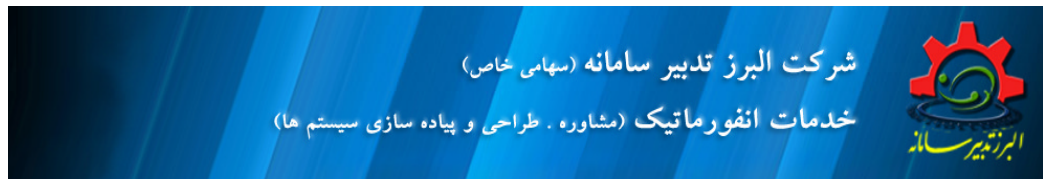
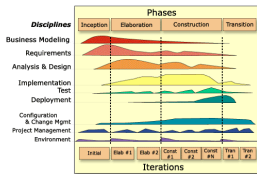
- تبدیل نیازمندی‌ها به طراحی سیستم که قرار است بوجود آید .
- پیدایش یک معماری مستحکم برای سیستم
- سازگار ساختن طراحی برای هماهنگ شدن با محیط پیاده‌سازی و طراحی آن برای کارایی بهتر

در اوایل فاز Elaboration، بر ایجاد یک معماری ابتدایی برای سیستم تمرکز می‌شود، که یک معماری کاندیدا برای فراهم کردن یک نقطه‌ی شروع برای تحلیل اصلی ارائه شود. اگر معماری قبلا وجود دارد (یا بدلیل اینکه در تکرارهای قبلی، در پروژه‌های قبلی تولید شده یا از یک چارچوب کاربردی بدست آمده)، تمرکز کار برای اصلاح معماری، تحلیل رفتار و ایجاد یک مجموعه‌ی اولیه از عناصر است که رفتار مناسب را فراهم می‌آورند

پیاده‌سازی (Implementation)

اهداف پیاده‌سازی عبارتند از:

- تعریف سازمان کد، برحسب زیر مجموعه‌ای از مجموعه‌های پیاده‌سازی سازمان یافته در لایه‌ها
- پیاده‌سازی کلاس‌ها و اشیاء بوسیله مؤلفه‌ها (فایل‌های منبع، باینری‌ها، فایل‌های اجرایی و ...)
- تست اجزاء تولید شده به عنوان واحدها
- مجتمع‌سازی نتایج تولید شده توسط پیاده سازان فردی (یا تیم‌ها) به صورت یک سیستم قابل اجرا



- دیسیپلین پیاده‌سازی مرز خود با تست را به اینکه تک تک کلاس‌ها چگونه تست واحد می‌شوند، محدود می‌کند. تست سیستم و تست مجتمع سازی در دیسیپلین تست انجام می‌گیرد.

آزمون (Test)

دیسیپلین تست از بسیاری جهات مانند یک ارائه دهنده خدمات برای سایر دیسیپلین‌ها عمل می‌کند. تمرکز اولیه تست کردن بر بررسی و ارزیابی کیفیت‌های محقق شده از طریق کارهای زیر است:

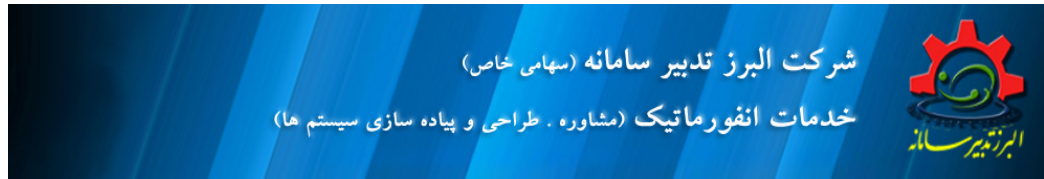
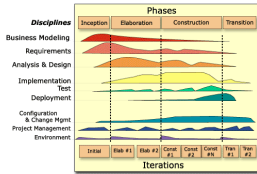
- یافتن و مستند کردن نقایص در کیفیت نرم‌افزار
- آگاهی دادن در مورد کیفیت نرم‌افزار بررسی شده
- اثبات اعتبار فرضیاتی که در طراحی و مشخصات نیازمندی‌ها ساخته شدند، از طریق نمایش‌های واقعی
- تصدیق عملکردهای محصول نرم‌افزار همانطور که طراحی شده است .
- تصدیق اینکه نیازمندی‌ها بدرستی پیاده‌سازی شده‌اند

یک تفاوت جالب ولی تاحدی ظریف میان دیسیپلین تست و سایر دیسیپلین‌ها در RUP این است که تست گرفتن، اساساً وظیفه‌ی یافتن و ارائه ضعف‌ها در محصول نرم‌افزار را داراست. برای اینکه این تلاش موفقیت‌آمیز باشد، لازم است از یک روش نسبتاً منفی و مخرب استفاده شود تا روشی سازنده. مسأله‌ای که بسیار حائز اهمیت می‌باشد این است که از دو روش اجتناب کنیم: یکی روشی که بطور مناسب و موثر نرم‌افزار را بکار نگیرد و مشکلات و ضعف‌های آن را نشان ندهد و دیگری روشی که آنقدر مخرب است که احتمالاً هیچگاه کیفیت محصول نرم‌افزاری را قابل قبول در نظر نمی‌گیرد.

استقرار (Deployment)

دیسیپلین استقرار فعالیت‌هایی را توضیح می‌دهد که تضمین می‌کنند محصول نرم‌افزاری برای کاربران نهایی‌اش در دسترس می‌باشد. دیسیپلین استقرار سه حالت استقرار محصول را توضیح می‌دهد.

- نصب اختصاصی



- آماده فروش کردن محصول نهایی
- دستیابی به نرم افزار از طریق اینترنت

در هر نمونه، تأکید روی تست محصول در سایت تولید است و سپس انجام تست بتا، پیش از اینکه محصول نهایتاً به مشتری تحویل داده شود. گرچه فعالیت‌های استقرار در فاز Transition به انتها درجه‌ی خود می‌رسند، اما برخی از فعالیت‌ها در فازهای قبلی برای طرح‌ریزی و آمادگی جهت استقرار انجام می‌شوند.

محیط (Environment)

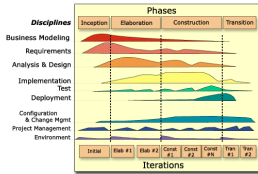
دیسپلین محیط بر فعالیت‌هایی که برای پیکربندی فرآیند برای یک پروژه لازم و ضروری‌اند، متمرکز می‌شود. این دیسپلین فعالیت‌های مورد نیاز برای تولید رهنمودهایی که در جهت پشتیبانی از یک پروژه لازم می‌باشند را توضیح می‌دهد. هدف فعالیت‌هایی محیطی فراهم آوردن محیط تولید (هم فرآیندها و هم ابزاری که تیم تولید را پشتیبانی می‌کنند) برای سازمان تولید کننده نرم‌افزار می‌باشد. جعبه ابزار مهندس فرآیند پشتیبانی ابزاری را برای پیکربندی یک فرآیند فراهم می‌کند. این مورد شامل ابزارها و نمونه‌هایی برای ایجاد سایت‌های وب پروژه و سازمان بر اساس RUP می‌شود.

مدیریت پروژه (Project Management)

مدیریت پروژه نرم‌افزاری، هنر متوازن ساختن اهداف متقابل، مدیریت ریسک و غلبه بر محدودیت‌ها برای تحویل موفقیت آمیز محصولی است که هم نیازهای مشتریان (کسانی که برای سیستم پول می‌پردازند) و هم نیازهای کاربران را برآورده کند. این حقیقت که پروژه‌های بسیار کمی هستند که واقعاً موفقیت‌آمیزند برای توضیح سخت بودن این کار، کافی می‌باشد.

اهداف این دیسپلین عبارتند از:

- فراهم کردن یک چارچوب برای مدیریت پروژه‌های صرفاً نرم‌افزاری
- فراهم کردن رهنمودهای عملی برای طرح‌ریزی، تعیین نیروی انسانی، اجرا و نظارت بر پروژه‌ها



- فراهم کردن یک چارچوب برای مدیریت ریسک

با این وجود، این دیسیپلین از RUP برای پوشش دادن همه‌ی جنبه‌های مدیریت پروژه نیست. برای مثال این دیسیپلین موارد زیر را پوشش نمی‌دهد:

- مدیریت افراد: استخدام، آموزش، رهبری
- مدیریت بودجه: تعیین، تخصیص و غیره
- مدیریت قراردادها: با پشتیبانی کنندگان و مشتریان

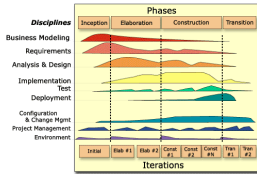
این دیسیپلین بطور عمده روی جنبه‌های مهم یک فرآیند تکراری تمرکز می‌کند که عبارتند از:

- مدیریت ریسک
- طرح ریزی برای یک پروژه‌ی تکراری، از طریق چرخه‌ی حیات و برای یک تکرار بخصوص
- **مدیریت پیکربندی و تغییرات (Configuration & Change Management)**
- نظارت بر پیشرفت یک پروژه‌ی تکراری و متریک‌ها

برای تأویل و تفسیر مدل بلوغ قابلیت انستیتو مهندسی نرم‌افزار (SEI CMM)، مدیریت پیکربندی و درخواست تغییر، تغییرات را به سمت خروجی‌های یک پروژه کنترل می‌کند و همچنین صحت و تمامیت خروجی‌های پروژه را حفظ می‌کند.

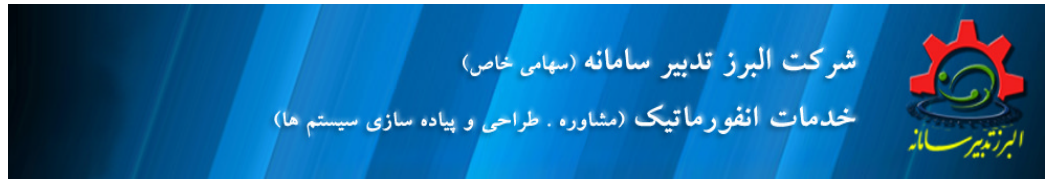
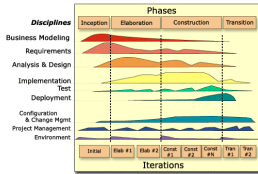
مدیریت پیکربندی و درخواست تغییر (CRM, CM) شامل موارد زیر می‌باشند:

- تشخیص موارد پیکربندی
- محدود کردن تغییرات آن موارد
- رسیدگی به تغییراتی که برای آن موارد ساخته شده
- تعریف و مدیریت پیکربندی آن موارد



جدول متدلوژی بومی شده برای فرآورده های پروژه به همراه الگوها و تشریح آنها

| فاز | | | | نقش | | فرآورده/تمویل دادنی (Artifacts) | Discipline (Discipline) |
|------|------|------|------|--------|----------------------|--|---|
| Trn. | Cns. | Elb. | Inc. | همکاری | مسئولیت | | |
| | | U | C | --- | طراحی | چشم انداز (Vision) | نیازمندی ها (Requirements) |
| U | U | U | C | --- | طراحی | واژه نامه (Glossary) | |
| | | U | C | --- | طراحی | مشخصات نیازمندی های نرم افزار (Software Requirements Specification) | |
| | | U | C | --- | طراحی | مدل موارد کاربرد سیستم (Use Case Model) | |
| | | U | C | --- | طراحی | گزارش عملگر (Actor Report) | |
| | | U | C | --- | طراحی | مشخصات مورد کاربرد (Use Case Specification) | |
| | U | C | | | نرم افزار | معماری نرم افزار (Software Architecture Document) | آنالیز و طراحی (Analysis & Design) |
| | U | C | | | نرم افزار | دیاگرام کلاس (Class Diagram) | |
| | U | C | | | نرم افزار | دیاگرام فعالیت (Activity Diagram) | |
| | U | C | | | نرم افزار | نمونه واسط کاربری (User Interface) | |
| | U | C | | | نرم افزار | مدل فیزیکی داده (Physical Data Model) | |
| U | C | | | | طراحی | نسخه اجرایی نرم افزار (Build) | پیاده سازی (Implementation) |
| U | C | | | | طراحی | مدل پیاده سازی (Implementation Model) | |
| | | C | | | طراحی / نرم افزار | طرح تست (Test Plan) | تست (Test) |
| | | C | | | طراحی / نرم افزار | تست (Test Case) | |
| U | C | | | | طراحی | مدل استقرار (Deployment Model) | استقرار (Deployment) |
| U | C | | | | طراحی | موارد لازم برای پشتیبانی کاربر نهایی (End User Support Material) | |
| | | U | C | --- | مدیر پروژه | طرح مدیریت پیکربندی (Configuration Management Plan) | مدیریت پیکربندی (Configuration Management) |



| | | | | | | | |
|---|---|---|---|-----|------------|--|--------------------------------------|
| U | U | U | C | --- | مدیر پروژه | سند تعریف نرم افزار (Software Definition Document) | مدیریت پروژه (Project Management) |
|---|---|---|---|-----|------------|--|--------------------------------------|

Inc.:Inception (آغازین)

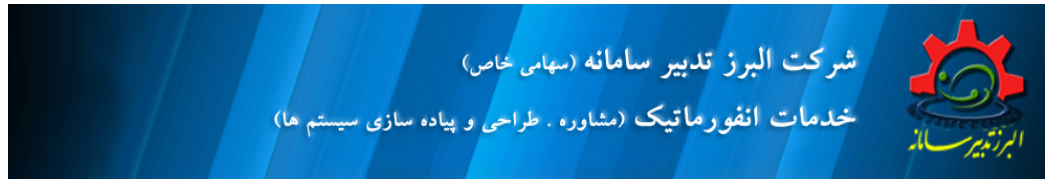
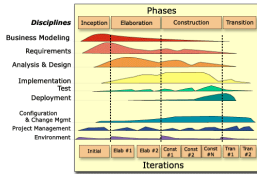
Elb.:Elaboration (تفصیل)

Cns.:Construction (ساخت)

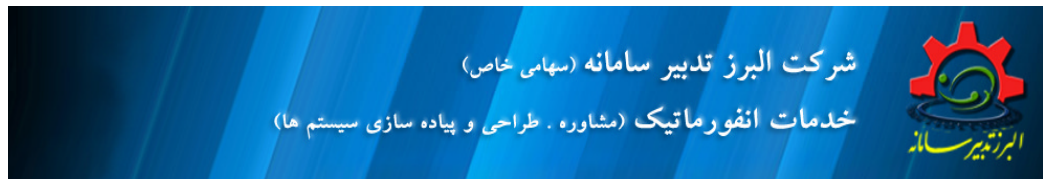
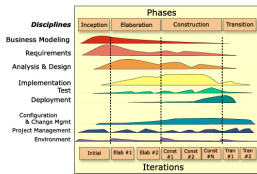
Trn.: Transition (انتقال)

در زیر هر یک از فرآورده های آمده در جدول برای مشخص شدن محدوده محتویات آنها شرح داده شده اند:

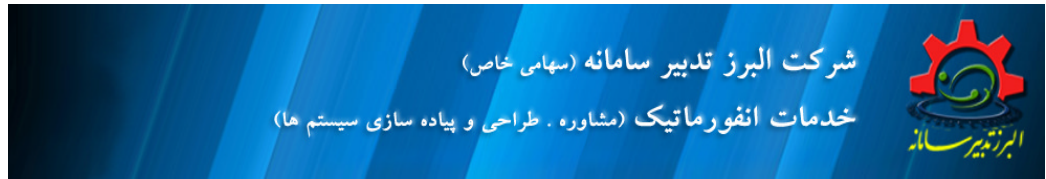
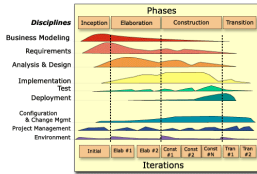
| شرح | فرآورده/تمویل دادنی (Artifacts) | ردیف |
|---|---------------------------------|------|
| این سند نیازمندی های کلان سیستم به ویژه از دید صاحبان سیستم را مشخص می کند. مهمترین مواردی که در این سند خواهد آمد عبارتند از: <ul style="list-style-type: none"> • شرح کاربران و صاحبان سیستم • مرور کلی محصول • قابلیت های محصول • محدودیت های سیستم • محدوده کیفیت سیستم • نیازمندی های جانبی سیستم • نیازمندی های مستند سازی | چشم انداز | ۱ |
| این سند تعاریف، اصطلاحات، اختصارات و تمامی واژه های کلیدی موجود در سیستم و مستندات آن را شرح می دهد. این سند به مرور زمان در طول کل پروژه تکمیل می گردد. | واژه نامه | ۲ |
| این سند تمام نیازمندی های نرم افزاری سیستم را مشخص کرده و تشریح می کند. مهمترین مواردی که در این سند خواهد آمد عبارتند از: <ul style="list-style-type: none"> • نیازمندی های کارکردی • نیازمندی های قابلیت استفاده، قابلیت اطمینان و قابلیت پشتیبانی • نیازمندی های کارآیی • محدودیت های طراحی | مشخصات نیازمندی های نرم افزار | ۳ |



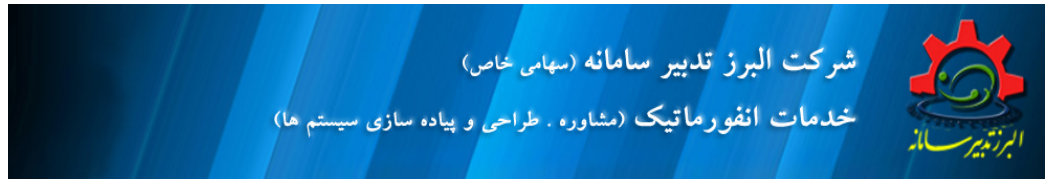
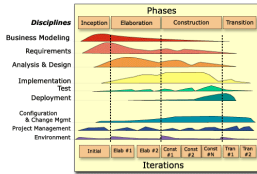
| | | |
|---|-----------------------------|----------|
| <ul style="list-style-type: none"> • نیازمندی های مستندسازی و Help • قطعات نرم افزاری مورد نیاز برای خرید • واسط های کاربری، نرم افزاری، سخت افزاری و ارتباطات • نیازمندی های مجوزها • استانداردها | | |
| <p>جزو سندهای پایه ای مورد استفاده در مراحل تجزیه و تحلیل، طراحی و تست سیستم هست که قابلیت های کارکردی سیستم را از دید کاربر خارجی به صورت مدل مورد کاربرد نشان می دهد. مهمترین مواردی که در این سند خواهد آمد عبارتند از:</p> <ul style="list-style-type: none"> • لیست موارد کاربرد و شرح مختصر آنها • ارتباطات میان روابط کاربرد • لیست عملگرها و شرح مختصر آنها • ارتباطات میان عملگرها • ارتباطات میان موارد کاربرد و عملگرها | <p>مدل مورد کاربرد</p> | <p>۴</p> |
| <p>این سند مجموعه ای از نقش ها که کاربران سیستم می توانند آنها را دارا باشند را مشخص کرده و تشریح می کند. مهمترین مواردی که در این سند خواهد آمد عبارتند از:</p> <ul style="list-style-type: none"> • شرح مختصر عملگرها • مشخصات عملگر • ارتباطات عملگرها | <p>گزارش عملگر</p> | <p>۷</p> |
| <p>این سند که به ازای یک مورد کاربرد نوشته می شود. مهمترین مواردی که در این سند خواهد آمد عبارتند از:</p> <ul style="list-style-type: none"> • شرح مختصر مورد کاربرد • جریان اصلی رخدادهای • جریان های فرعی رخدادهای • نیازمندی های ویژه مورد کاربرد • پیش شرط ها و پس شرط ها • ارتباطات مورد کاربرد • نمودارهایی برای تشریح فعالیت های مورد کاربرد مانند نمودار فعالیت (در صورت لزوم) | <p>مشخصات مورد کاربرد</p> | <p>۸</p> |
| <p>این سند از منظرهای مختلف، یک دید جامع از معماری سیستم را ارائه می کند. مهمترین مواردی که در این سند خواهد آمد عبارتند از:</p> | <p>سند معماری نرم افزار</p> | <p>۹</p> |



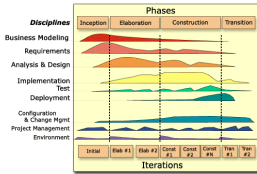
| | | |
|---|-------------------|----|
| <ul style="list-style-type: none"> • اهداف و محدودیت های معماری سیستم • معماری سیستم از دید موارد کاربرد • معماری سیستم از دید منطقی (زیرسیستم ها، Package ها، کلاس های مهم معماری نرم افزار، متدها و خواص مهم) • معماری سیستم از دید فرآیندها • معماری سیستم از دید استقرار سیستم • معماری سیستم از لحاظ پیاده سازی (لایه های نرم افزار) • معماری سیستم از دید اندازه و کارایی • معماری سیستم از دید کیفیت | | |
| <p>این سند طراحی نرم افزار را در قالب موارد ذیل شرح می دهد. مهمترین مواردی که در این سند خواهد آمد عبارتند از:</p> <ul style="list-style-type: none"> • ساختار Package طراحی اعم از زیرسیستم ها و Package های هریک از زیرسیستم ها و تشریح کلاسهای موجود در آنها • نمایش نحوه تعامل وابستگی و ارث بری کلاس ها • نمودار مولفه ها Component diagram | دیاگرام کلاس | ۱۰ |
| <p>در این سند برای هر یک از واسط های کاربری سیستم نمونه های آن با استفاده از ابزار گرافیکی تهیه می شود.</p> | نمونه واسط کاربری | ۱۱ |
| <p>این سند مدل فیزیکی داده هایی که در سیستم نگهداری می شود را نشان می دهد. اجزایی که مشخصات آنها در این سند خواهد آمد عبارتست از:</p> <ul style="list-style-type: none"> • مدل فیزیکی داده ها • کلاس های Persistence • ارتباطات • ایندکس ها • محدودیت ها • Trigger, Stored Procedure, View و ... • طرح هایی که روی سیستم مدیریت پایگاه داده تنظیم می شوند و ... | مدل داده | ۱۲ |
| <p>در این سند با ارائه مدل های استقرار نحوه قرارگیری ندهای سیستم نشان داده می شود که شامل موارد زیر خواهد بود</p> <ul style="list-style-type: none"> • ندهای پردازشی و مشخصات آنها • ارتباطات و مشخصات آنها • نحوه توزیع پردازشات و نرم افزارها بر روی ندهای پردازشی و مشخصات آنها | مدل استقرار | ۱۳ |



| | | |
|---|-------------------------------------|----|
| این فرآورده شامل نسخه ای اجرایی از سیستم است که قابلیت های نسخه نهایی محصول را به نمایش می گذارد. این فرآورده نسخه نهایی محصول نخواهد بود. | نسخه اجرایی نرم افزار | ۱۴ |
| این سند با ارائه مدل پیاده سازی سیستم، ترکیب فیزیکی پیاده سازی را نشان می دهد. مهمترین مواردی که در این سند خواهد آمد عبارتند از: <ul style="list-style-type: none"> • زیر سیستم ها • عناصر فیزیکی هر یک از زیر سیستم ها مانند شاخه ها، فایل ها، سورها • داده ها و فایل ها اجرایی و تشریح آنها • ارتباطات عناصر پیاده سازی | مدل پیاده سازی | ۱۵ |
| این سند اهداف و راهبردهای کلی تست سیستم را نشان می دهد. این سند در محدوده یک تکرار یا در محدوده کل پروژه تهیه می شود. مهمترین مواردی که در این سند خواهد آمد عبارتند از: <ul style="list-style-type: none"> • نیازمندیهای تست اعم از تست پایگاه داده، تست کارکرد، تست واسط های کاربری، تست کارایی، تست بار کاری و .. • استراتژی تست • تکنیک های تست ها • ابزار تست • منابع تست | طرح تست | ۱۶ |
| در این سند مشخصات یک مجموعه مشخصی از ورودی ها، حالات اجرا و خروجی های مورد نظر برای یک آیتم تست آورده می شود. مهمترین مواردی که در یک مورد تست خواهد آمد عبارتند از: <ul style="list-style-type: none"> • هدف و محدوده مورد تست • جزئیات محدوده تست • جزئیات راهبرد تست • ورودی ها، حالات اجرا • شرط موفقیت آمیز بودن یا نبودن تست | مورد تست | ۱۷ |
| مجموعه ای از اطلاعات بدست آمده از انجام موارد مختلف تست در این سند می آید. بر اساس اطلاعات بدست آمده از Log ها و درخواست های تغییر، ارزیابی تفصیلی برای هر یک از اهداف تست سیستم ارائه می شود این ارزیابی ها همراه با تجزیه و تحلیل و نتیجه گیری خواهد بود. | نتایج تست | ۱۸ |
| این سند مستندات و مواد لازم برای استفاده کاربران از سیستم را ارائه می کند. مهمترین مواردی که در این سند خواهد آمد عبارتند از: <ul style="list-style-type: none"> • سند راهنمای آموزشی کاربران | مواد لازم برای پشتیبانی کاربر نهایی | ۲۰ |



| | | |
|---|----------------------------|-----------|
| <ul style="list-style-type: none"> • فیلم و رسانه آموزشی کاربران • سند نصب و راه اندازی سیستم | | |
| <p>این سند اطلاعات لازم درباره فعالیت های مدیریت پیکربندی و کنترل را در طول چرخه حیات پروژه ارائه می کند. مهمترین مواردی که در این سند خواهد آمد عبارتند از:</p> <ul style="list-style-type: none"> • سازمان، مسئولیت ها و ارتباطات در مدیریت پیکربندی • ابزار مدیریت پیکربندی • محیط و فراساختار مدیریت پیکربندی • برنامه و متدهای لازم برای کنترل تغییرات پیکربندی | <p>طرح مدیریت پیکربندی</p> | <p>۲۱</p> |
| <p>مخزن یا Repository پروژه که شامل تمامی نسخه های فایل ها و شاخه های پروژه به عنوان یکی از خروجی های مدیریت پیکربندی تحویل خواهد شد.</p> | <p>مخزن پروژه</p> | <p>۲۲</p> |
| <p>این سند، سند پایه ای برای مدیریت پروژه می باشد. مهمترین مواردی که در این سند خواهد آمد عبارتند از:</p> <ul style="list-style-type: none"> • سازمان اجرایی پروژه، ارتباطات و مسئولیت ها • برنامه ریزی پروژه • منابع پروژه و بودجه • طرح تضمین کیفیت • طرح مدیریت ریسک ها • طرح اندازه گیری و ارزیابی • طرح پذیرش محصول • طرح حل مشکلات • طرح نظارت و کنترل پروژه • طرح مدیریت نیازمندی ها • طرح های فرآیندهای فنی پروژه • طرح مستندسازی • طرح فاز ها و تکرارهای پروژه، مشخصات و اهداف هر یک از آنها • مشخصات Release ها | <p>طرح توسعه نرم افزار</p> | <p>۲۳</p> |
| <p>در انتهای هر تکرار برای تکرار بعدی، برنامه ریزی تفصیلی آن در قالب طرح تکرار ارائه می شود. مهمترین مواردی که در این سند خواهد آمد عبارتند از:</p> <p>محدوده و اهداف تکرار</p> <p>برنامه ریزی تکرار</p> | <p>طرح تکرار</p> | <p>۲۴</p> |



شرکت البرز تدبیر سامانه (سهامی خاص)
 خدمات انفورماتیک (مشاوره . طراحی و پیاده سازی سیستم ها)

| | | |
|---|--|--|
| <p>منابع</p> <p>موارد کاربردی که در طرح پوشش داده می شوند</p> <p>شرط های ارزیابی که در انتهای تکرار باید آنها برآورده شده باشند</p> | | |
|---|--|--|