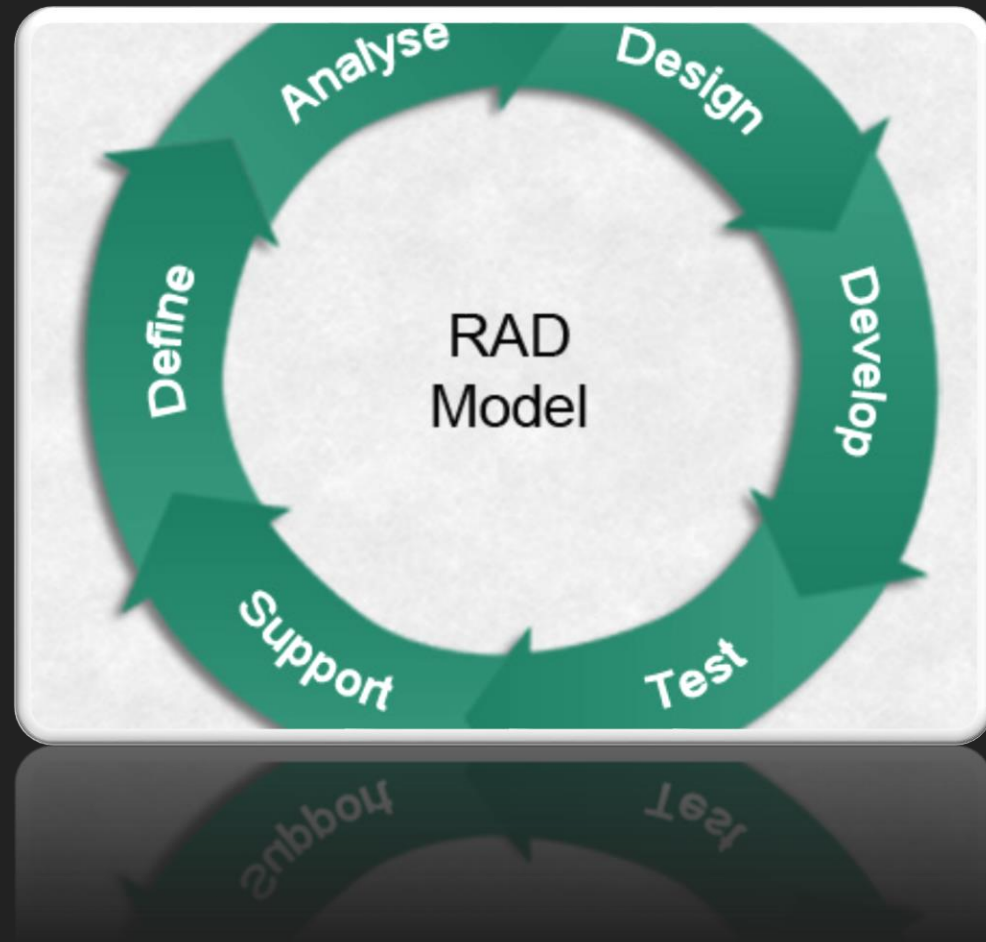
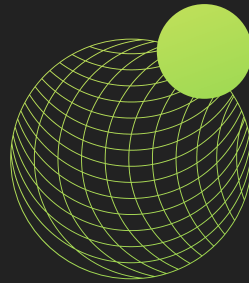


Rapid Development Methodologies



هدف:



سرعت بخشیدن به روند توسعه به منظور مقابله با
نیازهای تجاری که به سرعت در حال تغییر هستند

Rapid Development Methodologies

Rapid Application Development

محیط کسب و کار

و بنابراین به طور مداوم در حال
تغییر است

به طور فزاینده ای رقابتی

مشتری مدار

بین المللی

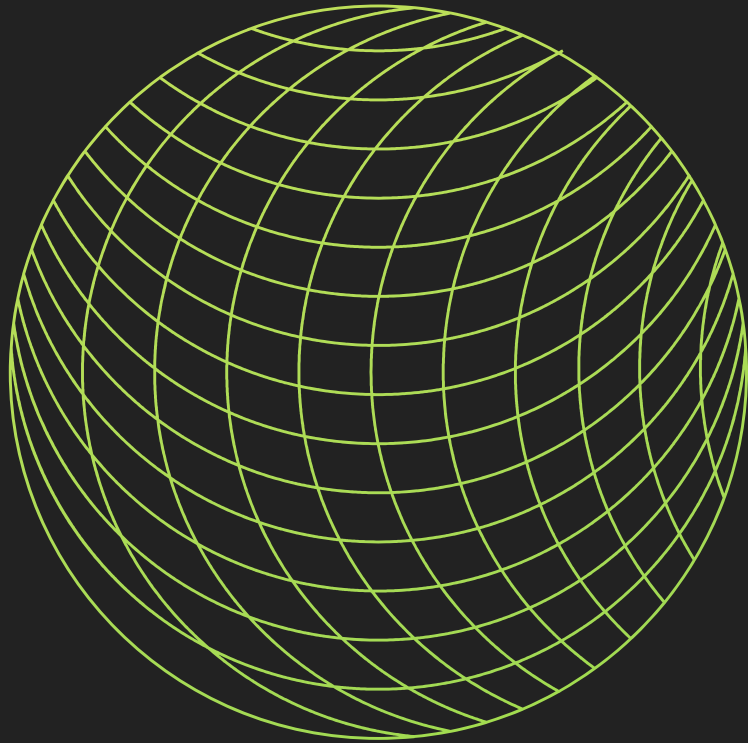
Rapid Application Development

عنوان یکی از الگوهای تولید نرم افزار می باشد، همچنین به عنوان نامی برای رویکرد جیمز مارتین برای توسعه سریع است

بطور کلی، رویکرد RAD برای توسعه نرم افزار، تاکید کمتری بر برنامه ریزی و تمرکز بیشتری بر فرآیند دارد. در قیاس با مدل آبشاری که در آن، تعریف دقیق مشخصات قبل از ورود به مرحله توسعه، خواسته می شود. رویکرد RAD تأکید بیشتری بر سازگاری و ضرورت تنظیم نیازمندی ها در پاسخ به دانش بدست آمده در پیشرفت پروژه می کند.



James Martin رویکرد RAD را به چهار فاز مجزا تقسیم کرده است:



فاز برنامه ریزی نیازمندی ها

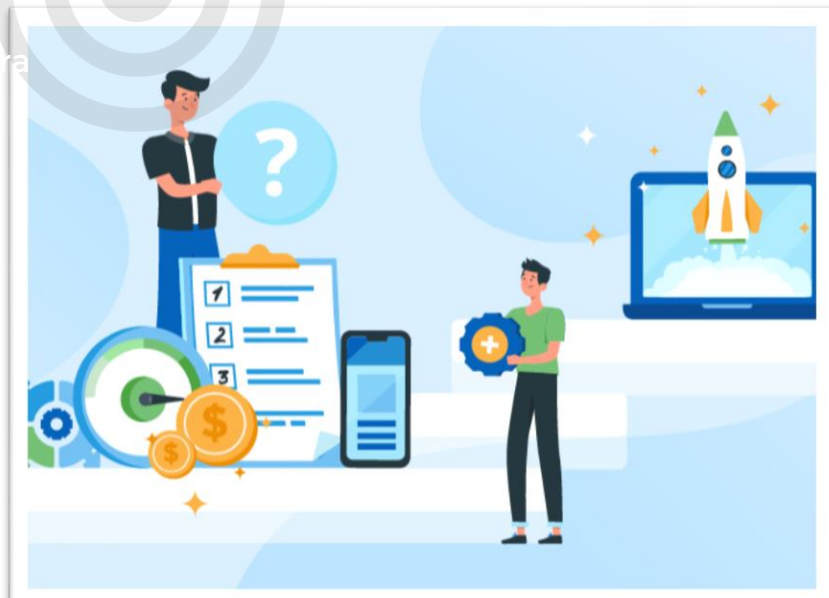
فاز طراحی کاربر

فاز ساخت

فاز تحویل

فاز برنامه‌ریزی نیازمندی‌ها

- ترکیبی از فاز برنامه‌ریزی سیستم و تحلیل سیستم در چرخه زندگی توسعه سیستم‌ها (SDLC) می‌باشد. کاربران و مدیران و کارکنان IT در خصوص نیازهای کسب و کار، محدوده پروژه، قیود و نیازمندی‌های سیستم بحث می‌کنند و به اتفاق نظر می‌رسند. این فاز زمانی به پایان می‌رسد که تیم در خصوص مسائل کلیدی به توافق برسند و تأیید مدیریتی را برای ادامه کار به دست بیاورند.
- در این فاز کاربران با تحلیلگران سیستم، مدل‌های توسعه و نمونه‌های اولیه را که بیانگر تمامی فرآیندهای سیستم، ورودی‌ها و خروجی‌ها است را تبیین می‌کنند. طراحی کاربر، یک فرآیند پیوسته تعاملی است که به کاربر این امکان را می‌دهد تا با درک، تبدیل و در مواردی بهبود یک مدل کاری این امکان را ایجاد کند که سیستمی پاسخگوی نیاز کاربران باشد.



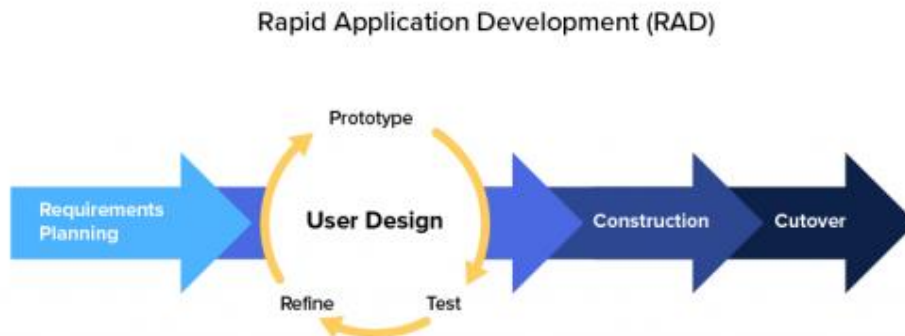
فاز ساخت

- در این فاز مشابه SDLC، تمرکز بر برنامه توسعه برنامه است اما کاربران همچنان مشارکت خواهند داشت و می‌توانند در ساخت صفحات واقعی و گزارش‌ها پیشنهادهایی برای تغییرات و بهینه‌سازی ارائه دهند. وظایف در این فاز عبارتند از برنامه‌نویسی، توسعه برنامه، کد نویسی، متصل کردن واحدها و تست کردن سیستم.
- شبیه کارهای نهایی در فاز استقرار SDLC شامل تبدیل داده‌ها، تست، مهاجرت به سیستم جدید و آموزش کاربر است. در مقایسه با مدل‌های سنتی، همه فرایند فشرده‌سازی شده‌است. در نتیجه، سیستم جدید خیلی زودتر توسعه داده شده، تحویل می‌شود و در جایگاه عملیاتی قرار می‌گیرد.



فاز ساخت

- بیشتر برای نمونه سازی یا Prototype استفاده می شود تا . Development
- اساس و پایه ی RAD استفاده از نمونه سازی سریع است.
- RAD یک تکنیک توسعه تیمی است که توسعه سیستم اطلاعاتی را تسریع می کند.
- RAD یک تکنولوژی کامل است که دارای ۴ فاز مشابه فاز های پروژه های سنتی است (planning , design , construction , cutover)
- RAD معمولاً از روش برنامه نویسی شی گرا استفاده می کند، که ذاتاً باعث استفاده مجدد از نرم افزار می شود. محبوب ترین زبان های برنامه نویسی شی گرا ، ++C و جاوا ، در بسته های برنامه نویسی بصری که اغلب به عنوان ارائه سریع برنامه های کاربردی توصیف می شود ، ارائه می شوند.



به صورت کلی (RAD توسعه سریع برنامه) شرایطی را فراهم می کند که محصولات با

سرعت بیشتر و با کیفیت بالاتر قابل توسعه باشند :

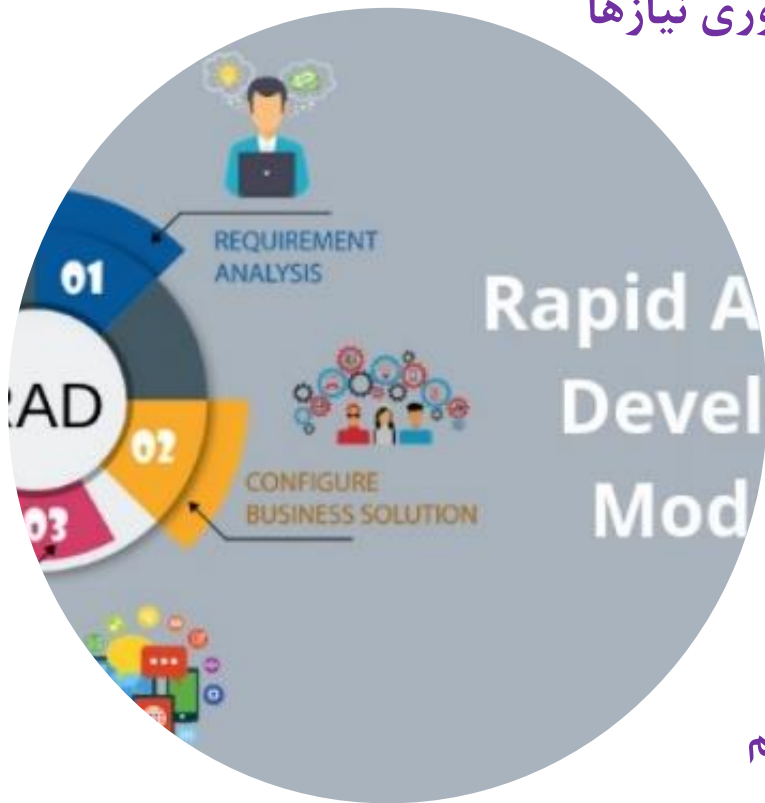
✓ استفاده از کارگاه ها یا گروه های متمرکز برای جمع آوری نیازها

✓ نمونه سازی و تست کاربر از طرح ها

✓ استفاده مجدد از مؤلفه های نرم افزار

✓ پیروی از برنامه ای که بهبود طراحی و رفع نواقص را تا نسخه بعدی به تعویق بیندازد .

✓ برگزاری جلسات بررسی و سایر ارتباطات غیر رسمی تیم



Methodologies

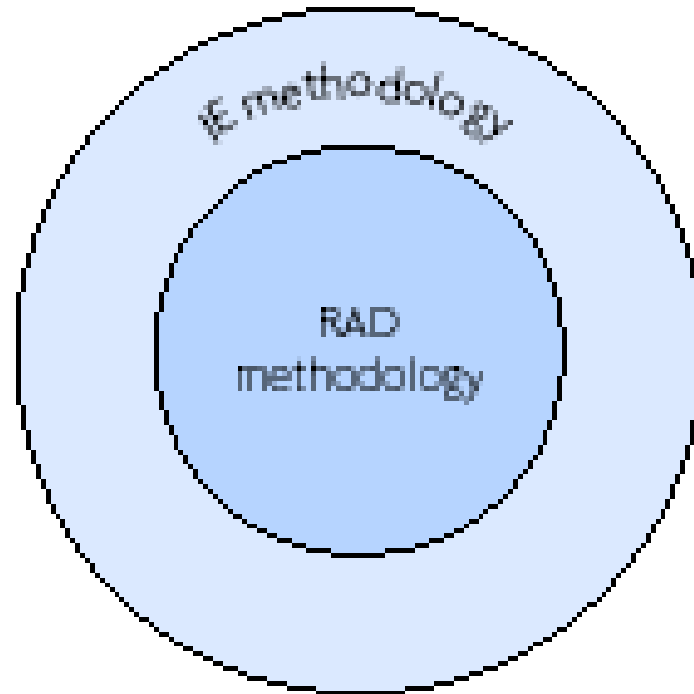
روش توسعه وب
WISDM

برنامه نویسی
افراطی **XP**

روش توسعه
سیستم پویا
DSDM

برنامه توسعه سریع جیمز
مارتین **jmRAD**

RAD and IE



RAD fits into the IE methodology

RAD مدل توسعه کاربردی سریع

تعاریف و مفاهیم:

در تعریف خلاصه ای از این مدل، یک مدل فرآیندی توسعه نرم افزار ترتیب خطی است که همان گونه که از نامش پیداست، قصد دارد فرآیند توسعه را تسریع کند. البته در تعاریف دیگر روش توسعه سریع (RAD) بیش از آنکه یک متدولوژی به معنای مصطلح آن باشد، همواره به عنوان بخش مکملی از برخی متدولوژیهای شناخته شده تلقی شده است. در واقع امروزه RAD را به عنوان یک تکنولوژی توسعه سیستمهای اطلاعاتی و به ویژه روشی برای تهیه نرم افزارهای اطلاعاتی می شناسند.

RAD مدل توسعه کاربردی سریع

RAD نخستین بار در چهارچوب متدولوژی مهندسی اطلاعات (IE) و توسط جیمز مارتین (مارتین، ۱۹۹۱) مطرح شد. مارتین در تحلیل خود از علل ناکامی رهیافت سنتی به توسعه سیستم ها (رهیافت متکی به SDLC و حتی روشهای ساخت یافته قدیمی) به این نکته توجه کرد که عوامل زیر در این ناکامی ها مؤثر بوده است:

الف) طولانی شدن روند توسعه

ب) عدم مشارکت کاربران نهایی

ج) بزرگی گروههای توسعه دهنده

RAD مدل توسعه کاربردی سریع

ایده RAD از تلاش برای غلبه بر این مشکلات پدید آمد.

به دلیل آنکه RAD بیش از آنکه یک راه حل تجویزی مشخص برای توسعه سیستم ها باشد، مجموعه ای است از راهکارهای عملی و در چهارچوب متدولوژی های مختلفی به کار بسته می شود، آن را نمی توان بسادگی جزء یکی از دو گروه متدولوژی های پردازش - مدار و یا داده - مدار طبقه بندی کرد. به همین خاطر شاید بهتر باشد RAD را یک متدولوژی مبتنی بر شبیه سازی با تأکید ویژه بر مشارکت کاربران نهایی به شمار آورد.

RAD مدل توسعه کاربردی سریع

در چهارچوب مفهومی RAD، هر سیستم اطلاعاتی بخشی است از منابع اطلاعاتی یک سازمان که در قالب یک ابزار (سیستم مکانیزه) توسط طیف وسیعی از کاربران، از مدیران ارشد گرفته تا کارمندان عملیاتی به کار گرفته می شود. به دلیل ماهیت نرم افزاری سیستم ها، هر سیستم اطلاعاتی را می توان مرکب از سه لایه زیر دانست :

لایه فنی (تکنولوژیکی) که از بستر سخت افزاری، ارتباطات و سطح پیاده سازی نرم افزاری (سیستم عامل، ابزارهای توسعه، RDBMS ها و ...) تشکیل می شود.

RAD مدل توسعه کاربردی سریع

لایه اطلاعاتی که مبتنی بر مدل‌های داده - فرآیند، بانک‌های اطلاعاتی و الگوهای دستیابی و استفاده از اطلاعات است .

لایه کاربردی که به واسطه سطح بالای سیستم - کاربر مربوط شده و کاربردهای عملیاتی، کنترلی، مدیریتی و تصمیم سازی را در برمی گیرد. در این لایه ها به ترتیب میزان دخالت و حساسیت خبرگان فنی کم شده و به میزان دخالت و حساسیت کاربران نهایی افزوده می شود.

RAD مدل توسعه کاربردی سریع

مطابق این الگو، به عنوان مثال در لایه فناوری، بیشترین نقش طراحی و تصمیم گیری بر عهده متخصصین خبرگان فنی است و کاربران نهایی یا در انتخاب عناصر این لایه نقشی ندارند و یا اینکه نقش آنها به صرف تأیید و کنترل محدود می گردد. از سوی دیگر توسعه سیستم در لایه کاربردها اساساً بر عهده کاربران نهایی است که با اعلام نیازها و تبدیل آنها به زبان سیستم، ابزار سیستم اطلاعاتی را به خدمت خود می گیرند.

RAD مدل توسعه کاربردی سریع

لایه میانی، یعنی لایه اطلاعات که الگوسازی داده ها و فرآیندها و کل عملیات سنتی تحلیل سیستم در آن صورت می گیرد، میدان عملیات مشترک خبرگان - کاربران است . در این لایه، توسعه سیستم ها باید با بده بستان مستمر اطلاعاتی میان کاربران و خبرگان صورت پذیرد. در هریک از این لایه ها، نوع اندرکنش بین این دو گروه چنین است :

RAD مدل توسعه کاربردی سریع

باید توجه کرد که این تبادل اطلاعات و مشارکت دو جانبه در هر لایه به صورت متقابل و در یک روند رفت و برگشت کامل شده و به صورت تدریجی نهایی می شود.

مراحل:

برای اجرای روش RAD نیازی به طی مراحل سنتی SDLC نیست، و به جای آن پیروی از یک الگوی چهار مرحله ای مطابق مراحل زیر توصیه می شود:

RAD مدل توسعه کاربردی سریع

برنامه ریزی نیازها

در روش RAD تأکید زیادی بر روش انجام مراحل ابتدایی توسعه سیستم (برنامه ریزی و امکان سنجی) صورت می گیرد. در این مراحل است که وظیفه اساسی تعریف نیازهای کاربران به انجام می رسد. روش اساسی توصیه شده برای این مرحله برگزاری جلسات برنامه ریزی مشترک نیازها (JRP) است. هدف از JRP ها تعیین نیازهای مدیریتی سطح بالا در مورد سیستم، در سطحی راهبردی و کلان است.

RAD مدل توسعه کاربردی سریع

طراحی توسط کاربر

تعیین و تدقیق نیازهای کاربر (به صورت جزئی)، تهیه مدل‌های داده و فرآیند (مدلسازی)، طراحی واسط‌ها، گزارش‌ها، صفحات و سایر عناصر سیستم از وظایف این مرحله است. روش اصلی انجام این مرحله برگزاری جلسات طراحی مشترک (JAD) است.

JAD یک جلسه برنامه ریزی شده است که با شرکت ترکیبی از کاربران عمده سیستم، مدیران و کارشناسان توسعه سیستم تشکیل می‌شود.

RAD مدل توسعه کاربردی سریع

ساخت

در مرحله ساخت طراحی انجام شده توسط کاربران در مرحله قبل، به طراحی تفصیلی و سپس به کد (برنامه) تبدیل می شود. این مرحله توسط گروههای کوچکی از کارشناسان طراح و برنامه نویس که به ابزارهای مناسب CASE مجهزند صورت می گیرد. هریک از این گروهها را یک گروه SWAT می نامند.

RAD مدل توسعه کاربردی سریع

بهره برداری

پس از تحویل برنامه های سیستم به کاربران، به همراه مستندات لازم، مرحله بهره برداری آغاز می شود. در این مرحله، اقدامات زیر صورت می گیرد :

- تحویل و نصب برنامه ها در محیط واقعی

- آموزش کاربران

- انجام تغییرات سازمانی لازم برای عملیاتی شدن سیستم

- اجرای همزمان سیستم های جدید و قدیم و انتقال تدریجی به سیستم

جدید

RAD مدل توسعه کاربردی سریع

ابزارها و روشهای نمودارسازی: به طور کلی استفاده از همه ابزارهای نمودارسازی توصیه شده در IE مانند (نمودارهای کنشی، DFD , ERD , Functional Decomposition) به ویژه در مراحل طراحی توسط کاربر و ساخت در این روش نیز مجاز است .

ابزارهای CASE : یکی از ویژگیهای ممیزه RAD تأکید شدید آن بر استفاده از ابزارهای CASE در جریان پیشبرد مراحل تحلیل، طراحی و ساخت سیستم است . استفاده از CASE به دو دلیل عمده زیر برای اجرای روش RAD ضروری است :

RAD مدل توسعه کاربردی سریع

- سرعت در تبدیل خواسته ها و نظرات کاربران به مستندات سیستم
- تبدیل سازگار و سریع مستندات طراحی سیستم به نمونه های قابل بازدید توسط کاربر

جلسات JAD: جلسات JAD به عنوان ابزار عمده پیشبرد در روند اجرای روش RAD هستند. جلسات JAD بیش از آنکه یک ابزار واحد باشند محمولی برای اجرای مجموعه ای از روشها و ابزارهای دیگرند. در حال حاضر JAD در بسیاری از متدولوژی های توسعه سیستم به عنوان یک روش سریع و مناسب برای دریافت نظرات کاربران، به عنوان جایگزینی برای روشهای سنتی مصاحبه و تکمیل پرسشنامه پذیرفته شده است.

RAD مدل توسعه کاربردی سریع

مارتین، همزمان با تلفیق تکنیک JAD در روش RAD دو عنصر مهم را به این جلسات افزود :

- استفاده از یک ابزار CASE مجتمع به عنوان کانون جلسه
 - تأکید بر نمونه سازی به عنوان روش مبادله افکار و نظرات در جلسات JAD
- کاربرد: امروزه محیط های برنامه نویسی دیداری که شامل ۳ تکنولوژی پیشرو: زبانهای برنامه نویسی نسل چهارم (GL4)، تکنیکهای شیء گرایى ، و روش RAD هستند به عنوان ایده مسلط بر بازار ابزارهای توسعه نرم افزار درآمده اند و نقش مهمی در همه گیر شدن و رواج این ابزارها ایفا می کنند.

RAD مدل توسعه کاربردی سریع

یک پایه این محیط ها ایده هایی است که در تکنولوژی RAD نقش اساسی داشته اند، یعنی لزوم مشارکت عملی کاربران نهایی در جریان توسعه سیستم، نمونه سازی سریع، تلفیق CASE با ابزارهای برنامه نویسی و ... به همین دلیل کل محیط های دیداری را می توان از کاربردهای غیر مستقیم RAD دانست .

RAD مدل توسعه کاربردی سریع

مزایا و معایب:

مزایا:

- چون RAD با هدف سرعت بخشیدن به روند توسعه سیستم ها طراحی شده است، بدیهی است که مزیت عمده آن سرعت بالای تهیه سیستم است. به این ترتیب همه اشکالاتی که از طولانی شدن زمان توسعه سیستم ناشی می شود (هزینه های بالا، تغییر سازمان و در نتیجه تغییر خواسته ها، کاهش حمایت مدیریت و ...) با این روش مرتفع می شود.

RAD مدل توسعه کاربردی سریع

- مراحل RAD به گونه ای طراحی شده است که در تمام مراحل، مشارکت عملی کاربران نهایی را در تصمیم گیری و حتی طراحی سیستم تضمین کند.
- محدود بودن گروههای تولید کننده سیستم در روش RAD (بین ۳ تا ۴ نفر) ارتباطات لازم را برای هماهنگی و کنترل پروژه ها به میزان قابل توجهی کاهش می دهد و به همین دلیل راهبری و مدیریت پروژه های RAD را بسیار ساده تر می کند.

RAD مدل توسعه کاربردی سریع

- کاهش در زمان توسعه.
- افزایش قابلیت استفاده مجدد کامپوننت ها (اجزاء).
- بررسی های اولیه سریع رخ می دهند.
- دریافت بازخورد استفاده کنندگان رو با روی باز می پذیره.
- ادغام از همان ابتدا، بسیاری از مسائل ادغام رو حل می کنه.

RAD مدل توسعه کاربردی سریع

معایب:

- RAD اصولاً برای تهیه سیستم های کوچک و متوسط طراحی شده است . تعداد کاربران و گروههایی که از سیستم های بزرگ و بسیار بزرگ استفاده می کنند (مانند سیستم های بانکی توزیع شده، یا سیستم های کنترل صنعتی گسترده) آنچنان زیاد است که عملاً مشارکت دادن همه آنها در روند توسعه سیستم غیرممکن می شود

RAD مدل توسعه کاربردی سریع

- به قوی/ضعیف بودن تیم و اجراهای فردی اعضای تیم برای شناسایی نیازمندی های بیزینس وابسته هست.
- فقط سیستم هایی که پیمانه ای (ماژولار) هستند می تونم از مدل RAD استفاده کنن.
- مستلزم داشتن مهارت های زیاد توسعه و طراحی هست.
- وابستگی زیادی داره با مهارت های مدل سازی.
- از اونجایی که خیلی به مدل سازی نیاز داره برای پروژه های کوچک قابل استفاده (به صرفه) نیست.

RAD مدل توسعه کاربردی سریع

- به قوی/ضعیف بودن تیم و اجراهای فردی اعضای تیم برای شناسایی نیازمندی های بیزینس وابسته هست.
- فقط سیستم هایی که پیمانه ای (ماژولار) هستند می تونم از مدل RAD استفاده کنن.
- مستلزم داشتن مهارت های زیاد توسعه و طراحی هست.
- وابستگی زیادی داره با مهارت های مدل سازی.
- از اونجایی که خیلی به مدل سازی نیاز داره برای پروژه های کوچک قابل استفاده (به صرفه) نیست.

RAD مدل توسعه کاربردی سریع

- استفاده از ابزار اصلی RAD یعنی جلسات JAD در بسیاری از موارد دشوار یا حتی غیرممکن است . اولاً تفهیم کارآیی و ضرورت این روش به مدیران سازمان و سایر کاربران آسان نیست . ثانیاً مدیران ارشد و میانی معمولاً آنچنان درگیر فعالیت های اجرایی و روزمره هستند که انتزاع آنان از کارهای عادی حتی به مدت یک یا دو روز آنهم به صورت جمعی عملاً ناممکن می نماید.
- ثالثاً در کشورهایمانند کشور ما نباید انتظار داشت که همه مدیران شرکت کننده در JAD با مفاهیم و روشهای تهیه سیستم و به ویژه با ابزارهای مدلسازی آشنا باشند

RAD مدل توسعه کاربردی سریع

به همین دلیل مشارکت عملی اینان در جریان تحلیل و طراحی که در روش RAD حیاتی است، عملاً متحقق نخواهد شد.

- کارآیی RAD بیشتر در سازمانهای است که سابقه استفاده از سیستم های مکانیزه را دارند و فرهنگ استفاده از سیستم های اطلاعاتی تا حد زیادی در آنها رایج شده است . استفاده از RAD به ویژه در سازمانهایی که فاقد سیستم های مکانیزه قبلی هستند، دشوار است .

رسیدن به هدف برای توسعه سریع است

James Martin's RAD (JMRAD)

- خصوصیات عمومی
 - توسعه فزاینده و نمونه سازی
 - بازه زمانی (پیشنهادهای از ۹۰ روز تا ۶ ماه متفاوت است)
 - به تعدادی از اجزای تقسیم شده است که به ترتیب اولویت بندی شده (با توجه به توابع)
 - زمان و منابع ثابت هستند ، عملکردها متفاوت است (برخلاف توسعه سنتی)
- قانون ۸۰/۲۰
 - ۸۰٪ از ویژگیهای با ۲۰٪ از منابع قابل تحویل است
- قوانین MOSCOW
 - الزامات اولویت بندی می شوند M : باید داشته باشد S . باید داشته باشید C = می تواند داشته باشد W . نخواهد داشت
 - کارگاه های مشترک توسعه برنامه برای مشارکت کاربر
 - اسپانسرها و پشتیبانها مشخص کرده اند
 - ابزارهایی برای سرعت بخشیدن به روند توسعه

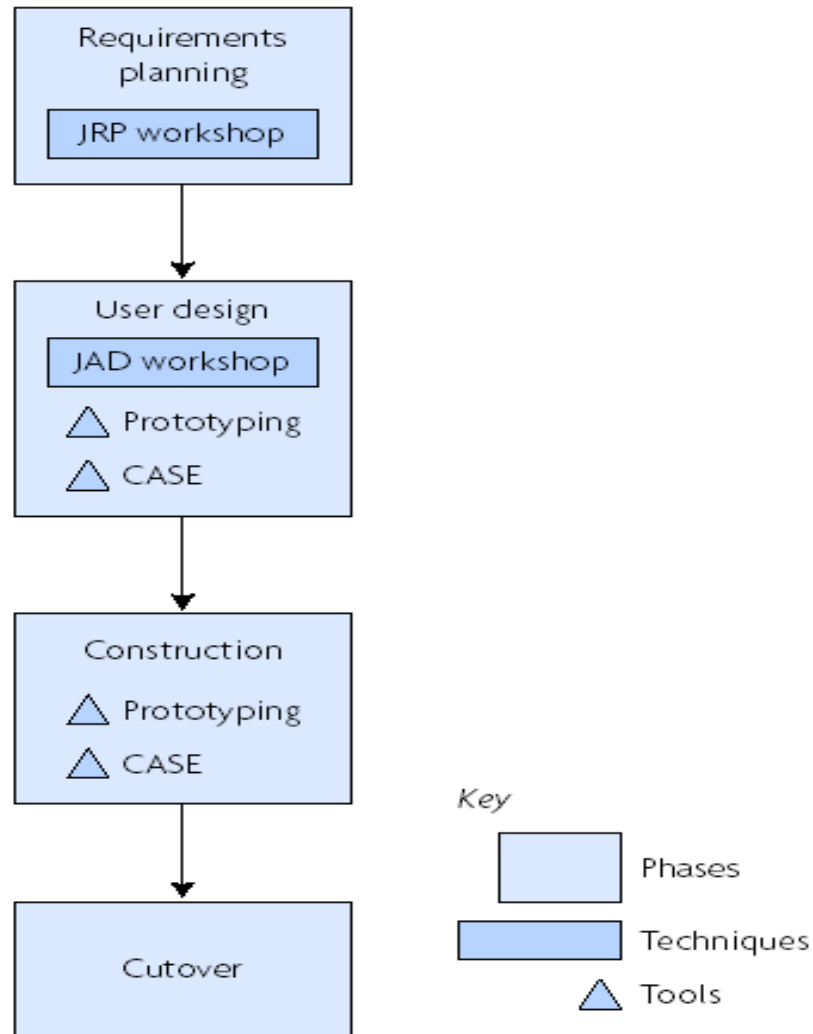


خصوصیات چهار فاز اصلی

- برنامه ریزی مورد نیاز (برنامه ریزی مشترک مورد نیاز)
- طراحی کاربر (توسعه برنامه های مشترک)
- ساخت و ساز (طراحی کاربر تا طراحی دقیق و برنامه نویسی بتن ؛ نمونه سازی اولیه)
- Cutover مقایسه بین سیستم قدیمی و جدید

- نمونه سازی / رویکرد تکاملی
- شناسایی و درگیر کردن کاربران در مراحل اولیه توسعه
- اخذ تعهد از کاربران تجاری
- به یک مجموعه ابزار با مخزن پیچیده نیاز دارد

مراحل رویکرد RAD



Dynamic Systems Development Method (DSDM)

روشی برای ساخت و نگهداری سیستم های مبتنی بر رایانه ، که ترکیبی از استفاده موثر از ابزارها و تکنیک ها ، نمونه سازی اولیه و جدول زمانی تحویل پروژه فشرده است

برای اصلاح تصویر RAD از "سریع و کثیف" به روش مناسب (۱۹۹۴) ، اما بیشتر شبیه یک چارچوب است تا یک روش

Dynamic Systems Development Method (DSDM)

- این فریمورک بیشتر از یک متدولوژی به نظر می رسد و بسیاری از جزئیات نحوه انجام کارها را به کاربر منتقل می کند
- DSDM نه اصل را تعریف می کند که برای موفقیت پروژه در حوزه توسعه سریع بسیار مهم است
- به نظر می رسد مفاهیم زیربنایی اساسی با مفاهیم آشنایی سیستم های سخت از امکان سنجی ، مفهوم عملیات ، مدل سازی عملکردی ، طراحی و اجرا پیوند نزدیک دارند.

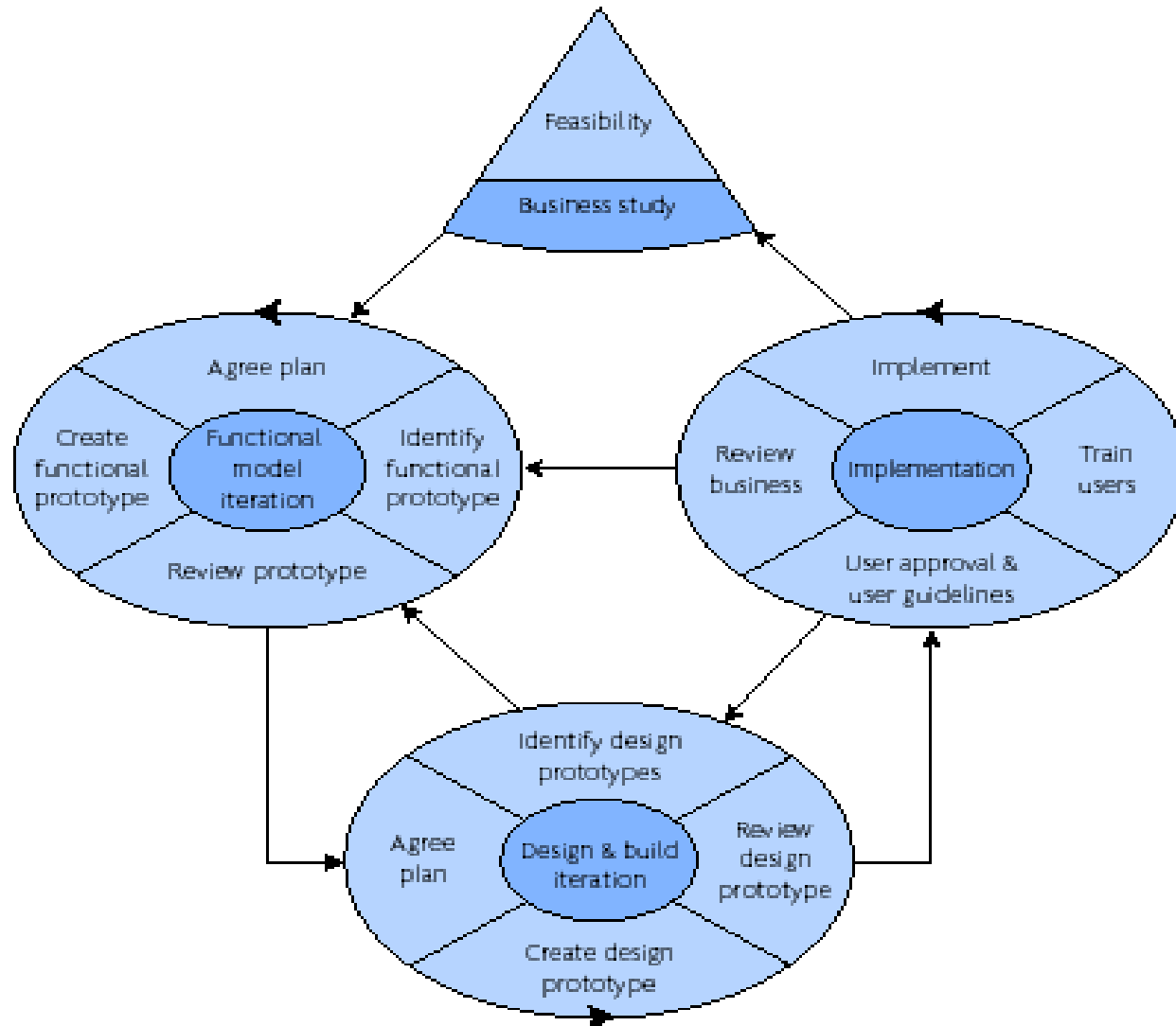
Dynamic Systems Development Method (DSDM)

۱. مشارکت فعال کاربر ضروری است
۲. تیم ها باید تصمیم بگیرند. چهار متغیر اصلی توانمندسازی عبارتند از:
اختیارات ، منابع ، اطلاعات و پاسخگویی
۳. تحویل مکرر محصولات ضروری است
۴. سازگاری برای مقاصد تجاری معیارهای اساسی پذیرش کالاهای قابل
تحویل است
۵. توسعه هم افزایی و افزایشی برای همگرایی در یک راه حل دقیق تجاری
لازم است

Dynamic Systems Development Method (DSDM)

۶. همه تغییرات در طول توسعه برگشت پذیر هستند ، به عنوان مثال در صورت بروز مشکل در مسیری خاص بیشتر پیش بروید ، به آخرین نقطه مطمئن یا مورد توافق عقب بروید و مسیر جدید را شروع کنید
۷. الزامات در سطح بالایی قرار می گیرند ، یعنی الزامات سطح بالای کسب و کار ، پس از توافق ، منجمد می شوند. این اساساً محدوده پروژه است
۸. کل چرخه باید تست انجام شود ، به جای اینکه فقط در انتها جایی که منجمد شده ، تست را انجام دهید.
۹. رویکرد همکاری بین همه ذینفعان ضروری است

DSDM three pizzas and a cheese diagram



DSDM development lifecycle – five phases

۱. بررسی امکان سنجی
۲. بررسی تجارت
۳. تکرار مدل عملکردی
۴. طراحی سیستم و تکرار ساخت
۵. پیاده سازی

DSDM and users

تعریف نقش در این فرایند تأکید شده است

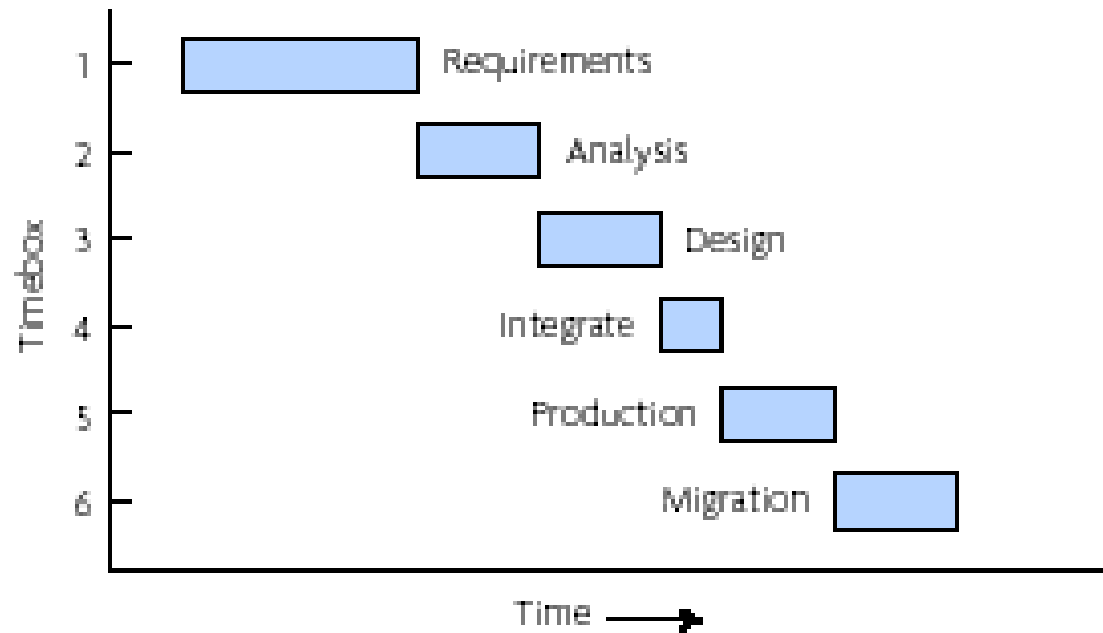
مدیر پروژه؛ تمام مهارت ها را باید داشته باشد اما تمرکز روی سرعت انجام عملیات باشد

سمت کاربر: کاربر سفیر (درک و شناخت نیازهای ارتباطی بین کاربران سازمان). کاربر بصیر (داشتن چشم انداز برای بدست آوردن سود برای یک سازمان)

سمت IT: نقش خاصی ندارد

حداقل یک نماینده از هر دو طرف باید تمام وقت حضور داشته باشد

Time boxes for DSDM project example





توسعه سریع برنامه یا همان Rapid Application Development (RAD) یک متودولوژی است که در سال ۱۹۸۰ پایه عرصه ظهور گذاشته است ولی مانند متودولوژی های دیگر مثل مدل آبشاری شناخته شده نمی باشد. این متودولوژی برخلاف مدل آبشاری می تواند ماهیت پویا داشته باشد و بر اساس نیاز مشتری تغییر نماید. به عبارت دیگر خود را با نیاز مشتری هماهنگ سازد.

ابزار های RAD یا Rapid Application Development به ابزارهایی گفته می شود که به فرد کمک می کند تا کلیه مراحل پیاده سازی را ببیند و prototype یا نمونه اولیه خود را ایجاد کند

در چند تکرار متفاوت تغییرات لازم را روی محصول خود ایجاد نماید، بدون اینکه نیاز باشد **core functionality** یا هسته اولیه عملیات در سیستم تغییر کند. این کار سبب می شود تا در توسعه یک نرم افزار نیاز نباشد هر بار از مرحله اسکچ اولیه، تغییرات رخ دهد و این نشان دهنده پذیرا بودن تغییرات در هر مرحله از توسعه می باشد. این ابزار و در واقع به بیان بهتر این متدولوژی کمک می کند تا ضعف موجود در مدل آبشاری پوشش داده شود.

در مدل آبشاری، کل روند به مراحل تقسیم می شد که در پایان هر مرحله و ورود به مرحله بعدی، امکان ایجاد تغییر در مراحل قبلی میسر نبود و یا به بیان بهتر هزینه این تغییرات در حدی است که کار را ناممکن می کند

حال آنکه در این مدل، حتی در مرحله تست، می توان قابلیت های جدیدی به سیستم اضافه کرد و این اتفاق با حداقل هزینه قابل انجام است. در این مدل تمرکز روی برنامه ریزی و تحلیل و حتی جزییات طراحی نیز گذاشته می شود در عین حال که رویکرد متدولوژی RAD سریع بودن و گرفتن فیدبک بین درخواست کننده و برنامه نویس جهت پیشبرد کار است. پروتوتایپ یا نمونه اولیه می تواند ابزاری باشد که به کمک آن بتوان به فهم مشترکی بین مشتری و تیم پیاده سازی رسید و از پرداخت زمان زیاد جلوگیری کرد.

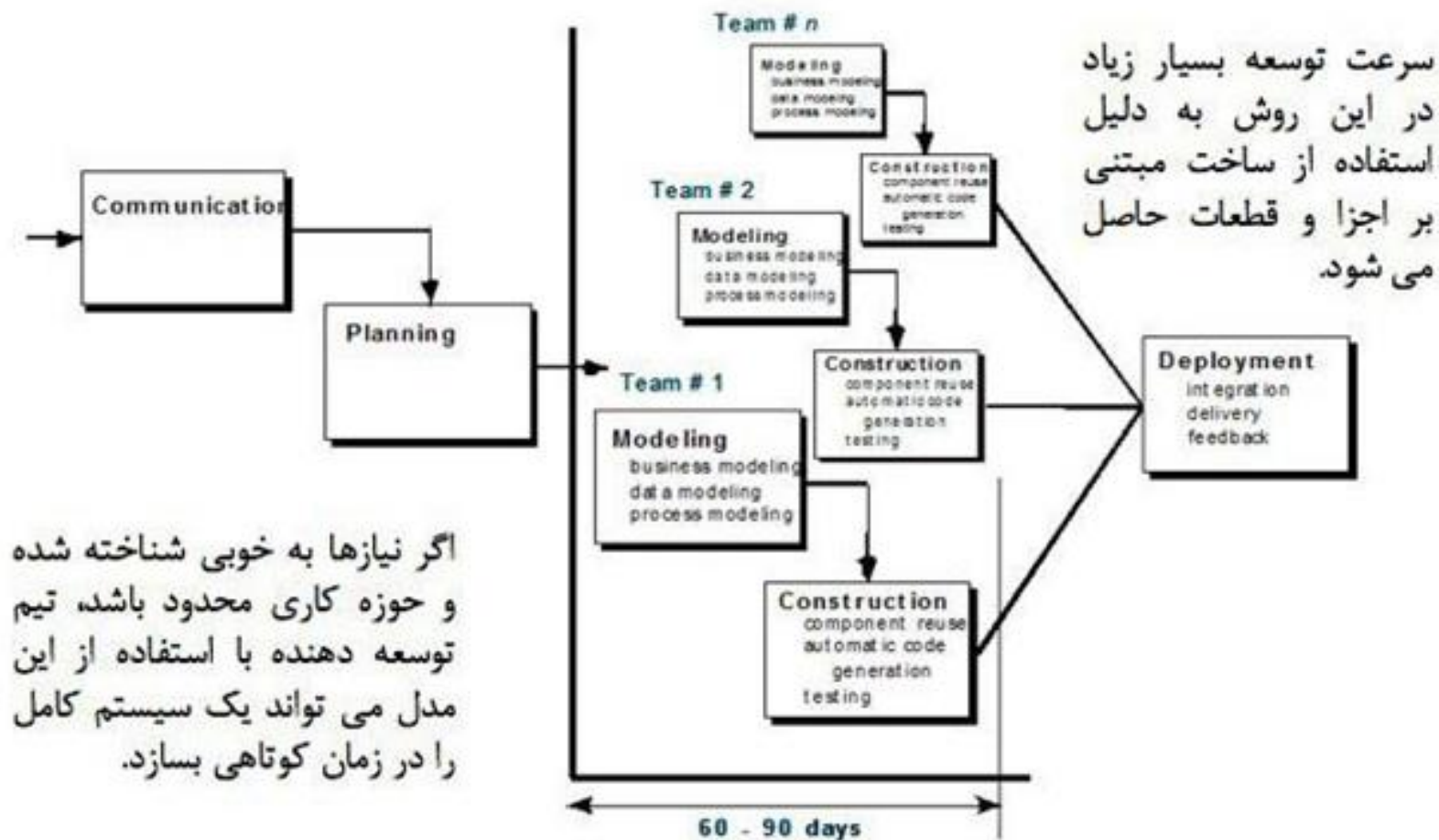
اگرچه مراحل این متدولوژی طی زمان تغییر کرده ولی در حالت کلی می توان این متدولوژی را در ۴ مرحله بیان کرد:

۱. تعریف نیازمندی
۲. ارائه پروتوتایپ یا نمونه اولیه
۳. گرفتن فیدبک
۴. نهایی کردن فرآیند

مدل نرم افزاری RAD به چه صورت است؟

در واقع RAD4 مدلی هست مبتنی بر دو مدل حلزونی و تکراری. در واقع یک مدل فرآیند تولید نرم افزار هست که تنها بر یک سیکل تولید بسیار کوتاه تأکید می کنه.

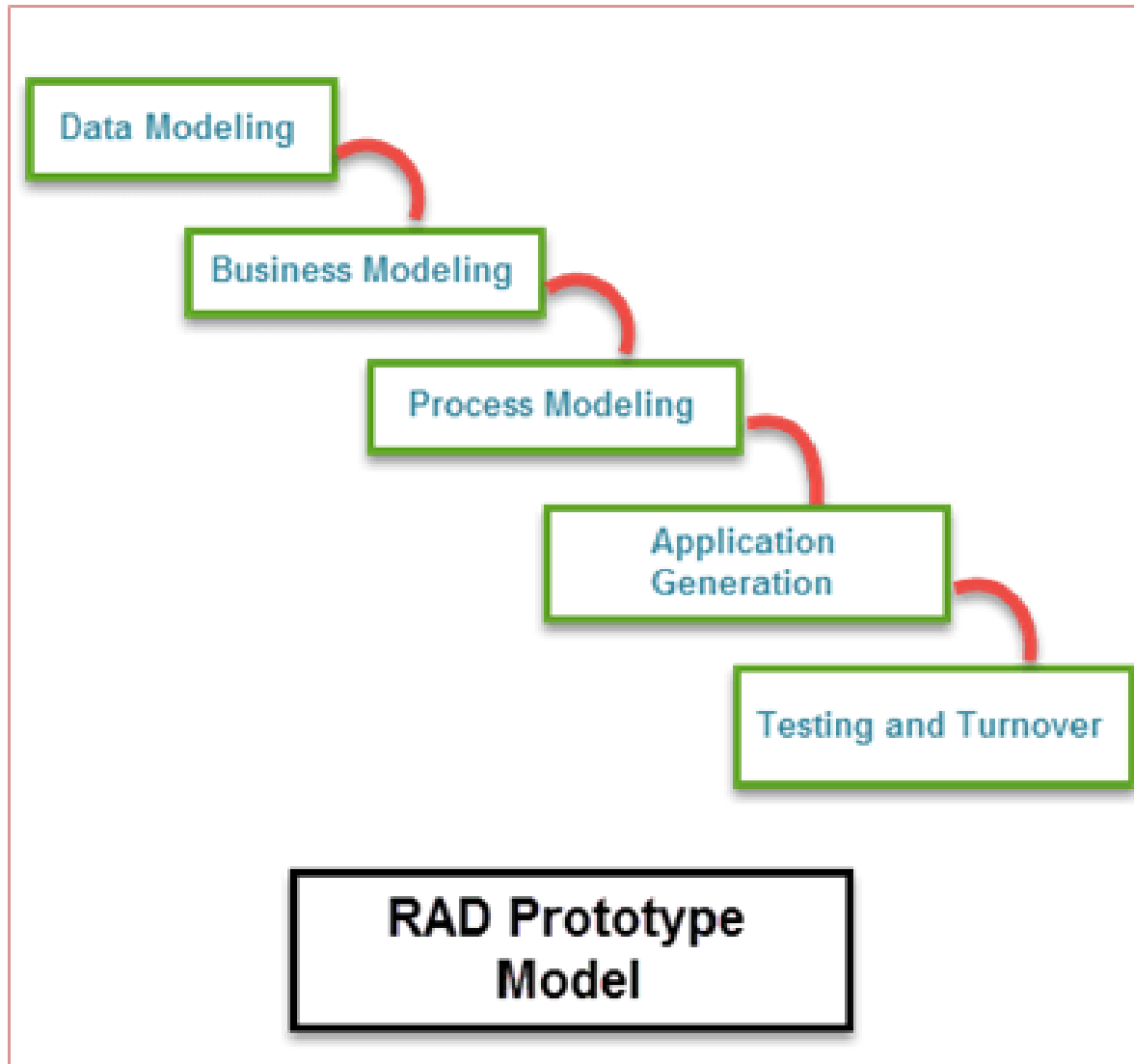
مدل نرم افزاری RAD به چه صورت است؟



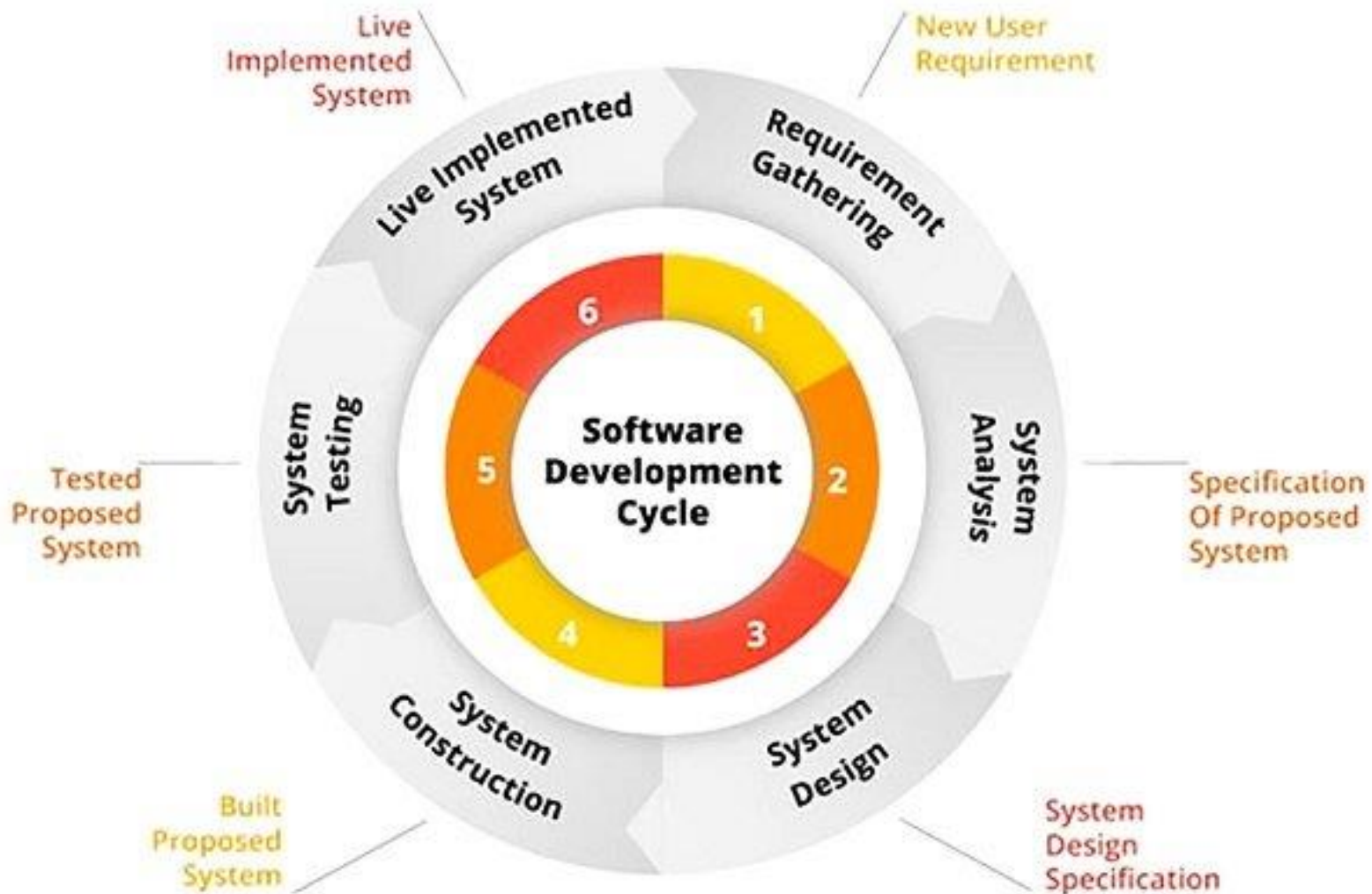
مدل نرم افزاری RAD به چه صورت است؟

مدل RAD برای ساخت سیستم های اطلاعاتی پیشنهاد میشه و کاربرد این مدل، تاکید بر کوتاه نمودن دوره ساخت نرم افزار هست. این مطلب بیانگر اینه که BPMS5 یک روش برآمده از اصول مهندسی است و تمام اصول مهندسی را برای اجرای یک فرآیند سریع و مطمئن در سازمان را رعایت کرده است.

مدل نرم افزاری RAD به چه صورت است؟



مدل نرم افزاری RAD به چه صورت است؟



Extreme Programming (XP)

- به خصوص برای برنامه های کوچک و متوسط مناسب است (زمانی که کل پروژه به ۳-۱۰ برنامه نویس نیاز دارد بهترین عملکرد را دارد)
- این یک سری اصول برای توسعه سریع نرم افزار است
- XP یک روش سبک وزن است
- چند قانون و روش
- توجه جزئی به اسناد و مدارک
- رضایت مشتری را تأکید می کند: این نرم افزار برای ارائه نرم افزار مورد نیاز مشتری در صورت نیاز طراحی شده است

Extreme Programming (XP)

- این توسعه دهندگان را قادر می سازد تا حتی در اواخر چرخه هم به تغییر نیازهای مشتری پاسخ دهند
- گزارشات کاربران
- لبه معماری
- برنامه نویسی زوجی
- برنامه ریزی
- طراحی
- در حال توسعه

Extreme Programming (XP)

- تولید (آزمایش)

XP: planning

- گزارشات مربوط به کاربران نوشته شود(سند)
- طرح انتشار (برنامه برای انتشارهای مکرر کوچک)
- سرعت یا روند اجرای پروژه را اندازه گیری کنید
- پروژه را به تکرار تقسیم کنید و هر کدام را با برنامه ریزی تکرار شروع کنید
- جلسات روزانه برای برقراری ارتباطات و یافتن مشکلات و راهحلهای آنها و ارتقا

XP: designing

- طراحی را ساده نگه دارید ؛ همیشه ساده ترین کاری را که می تواند کار کند انجام دهید و هرگز قبل از برنامه ریزی عملکرد به آن اضافه نکنید
- یک سیستم نام مناسب برای اشیا **objects** خود انتخاب کنید که همه بتوانند با آن ارتباط برقرار کنند
- استفاده از کلاس، مسئولیت ها و همکاری (CRC) کارت برای طراحی سیستم به عنوان یک تیم

XP: designing

- ایجاد خوشه های معماری: برنامه های ساده برای کشف راه حل بالقوه ، به عنوان مثال برنامه هایی که فقط به مسئله تحت بررسی می پردازند و سایر نگرانی ها را نادیده می گیرند
- چشم خود را بر روی نیازهای آینده و انعطاف پذیری بیشتر ببندید. فقط بر روی آنچه امروز برنامه ریزی شده تمرکز کنید
- طراحی های منسوخ شده را که درک ، نگهداری یا عملکرد بی استفاده آن بسیار دشوار است را دوباره فاکتور بندی کنید

XP: coding

- مشتری همچنان در مدت کد و آزمایش در دسترس هستند (از کارشناسان درخواست کنید نه از کارآموزان)
- کد باید مطابق با استانداردهای توافق شده نوشته شود
- ابتدا آزمون واحد را کدگذاری کنید (قبل از کد واقعی)
- برنامه نویسی زوجی: ۲ نفر در یک ایستگاه کاری کار می کنند
- ادغام کد پی در پی: فقط یک جفت از برنامه نویسان همزمان کد را ادغام می کنند

XP: coding

- ادغام کد اغلب در یک مخزن مشترک
- مالکیت کد جمعی تشویق می شود
- از بهینه سازی کد خودداری کنید
- از کار اضافی یا افزودن برنامه نویس خودداری کنید

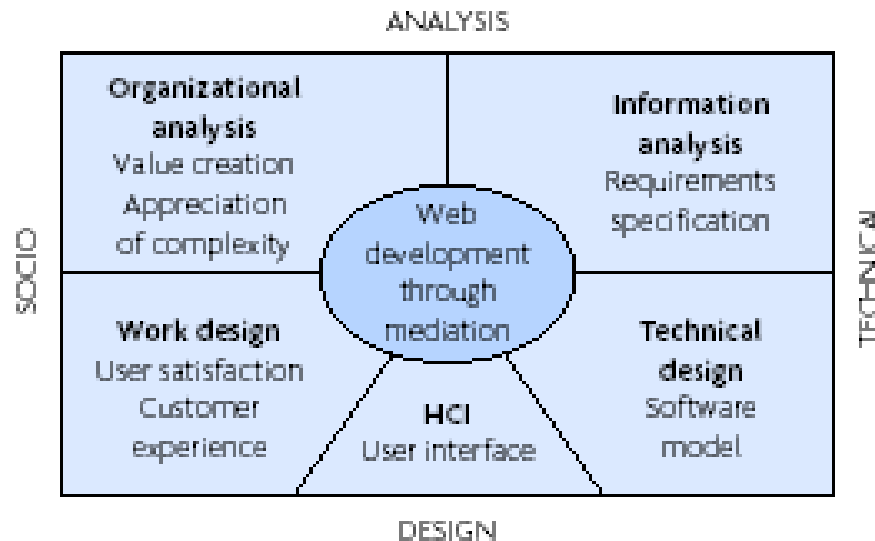
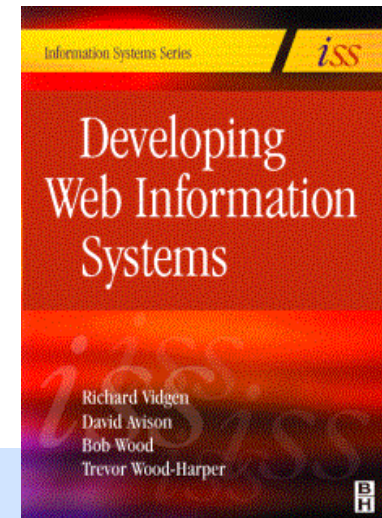
XP: testing

- همه کدها باید دارای تست واحد باشند
- قبل از انتشار ، همه کدها باید تمام آزمایشات واحد را پشت سر بگذارند
- هنگامی که یک اشکال یافت می شود آزمایشات بیشتری ایجاد می شود
- تست های پذیرش (ایجاد شده از گزارش کاربر در هر تکرار) اجرا می شوند و شماره جدید منتشر می شود

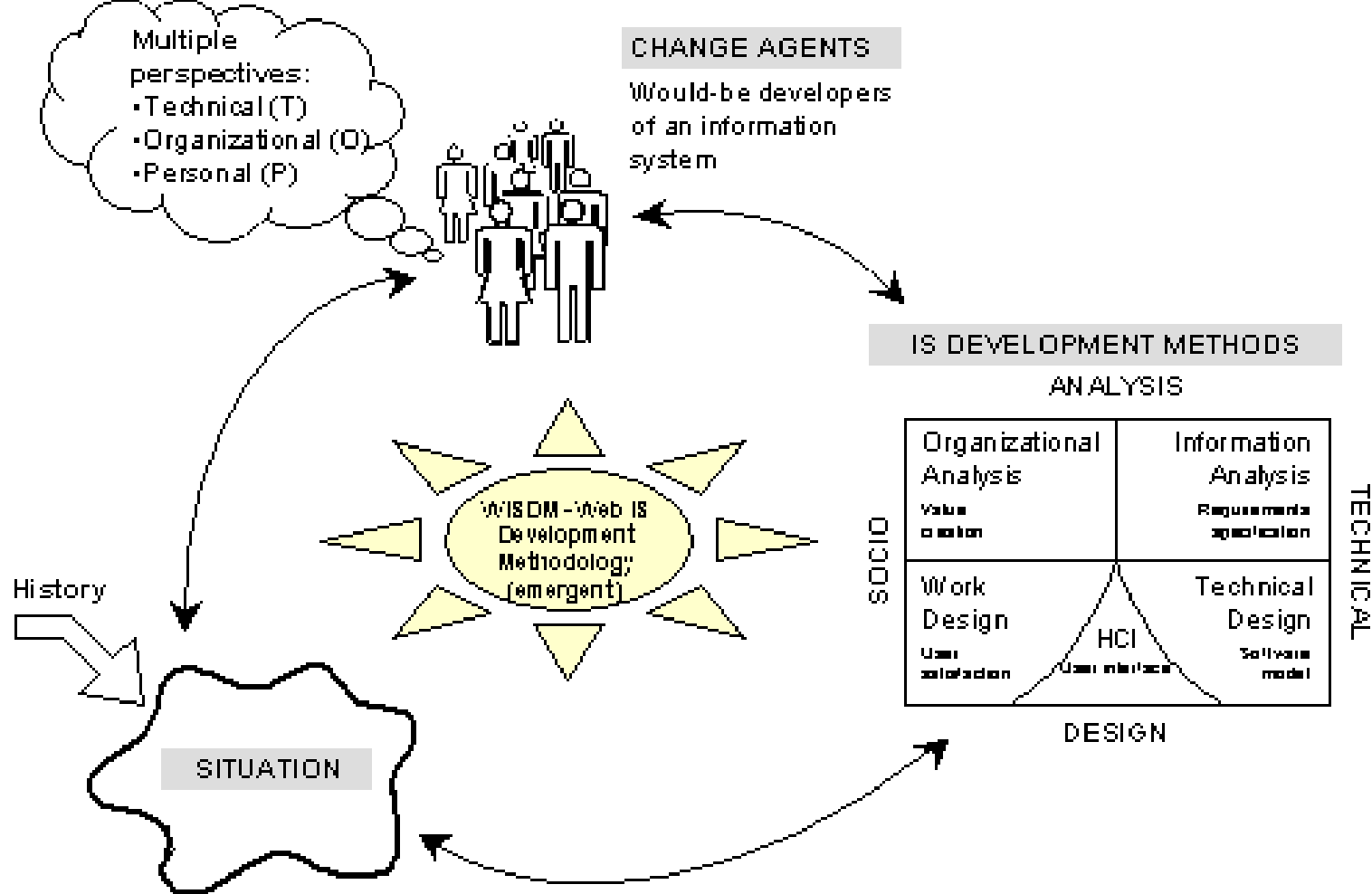
Web IS Development Methodology (WISDM)

WISDM یک چارچوب و روش برای توسعه سیستم های اطلاعاتی مبتنی بر وب است. این چارچوب تشخیص می دهد که یک روش در عمل از سه گانه وضعیت ، عوامل انسانی و روش ها بیرون می آید

Web IS Development Methodology (WISDM)



Reference: <http://www.wisdm.net/wisdm/index.htm>



Indicative methods for WISDM include:

Organizational Analysis: Soft Systems Methodology (SSM), e-business strategy

Information Analysis: Unified Modeling Language (UML)

Work Design: ETHICS, Participative Design, Web Quality (WebQual)

Technical Design: UML plus physical design as indicated by the target implementation environment

WebQual question

Usability

1. I find the site easy to learn to operate
2. My interaction with the site is clear and understandable
3. I find the site easy to navigate
4. I find the site easy to use
5. The site has an attractive appearance
6. The design is appropriate to the type of site
7. The site conveys a sense of competency
8. The site creates a positive experience for me

Information

9. Provides accurate information
10. Provides believable information
11. Provides timely information
12. Provides relevant information
13. Provides easy to understand information
14. Provides information at the right level of detail
15. Presents the information in an appropriate format

Interaction

16. Has a good reputation
17. It feels safe to complete transactions
18. My personal information feels secure
19. Creates a sense of personalization
20. Conveys a sense of community
21. Makes it easy to communicate with the organization
22. I feel confident that goods/services will be delivered as promised

Convergence

23. What is your overall view of this website?