بسم الله الرحمن الرحیم

دانشگاه صنعتی شریف

متدولوژی های ایجاد نرم افزار

استاد: دکتر رامسین

کیمیا علوی راد ۴۰۰۲۱۱۴۰۵

تمرین دوم

زمستان ۱۴۰۰

فهرست

فهرست اشکال

۱- متدولوژی USDP (UP)

متدولوژی UP در سال ۱۹۹۹ توسط Rambaugh ، Booch و Jacobson که توسعه دهندگان متدولوژی های OMT ، Booch و OOSE بودند، ساخته شد. این متدولوژی در واقع نسخه ساده شده RUP است. چرخه عمر[۱] این متدولوژی حاوی ۴ فاز inception، elaboration ، construction و transition می باشد که در هرکدام از این فازها iteration هایی انجام می شوند. هر iteration شامل ۴ جریان کاری[۲] است که عبارتند از نیازمندی ها[۳]، تحلیل[۴]، طراحی[۵]، اجرا[۶] و تست[۷]. در فاز inception ، چشم انداز، حوزه[۸] و توجیه اقتصادی[۹] پروژه تعریف می شود. در فاز elaboration تعریف محصول، اصلاح می شود؛ معماری مبنا[۱۰] تعریف می شود؛ و یک برنامه دقیق تر برای توسعه و استقرار ایجاد می شود. در فاز construction محصول تا حدی که قابل تحویل به کاربر باشد ساخته می‌شود. و در فاز transition انتقال محصول به محیط کاربر صورت می گیرد که شامل ساخت، تحویل، آموزش و برنامه ریزی برای پشتیبانی و نگهداری محصول می باشد. در ادامه به توضیح دقیق تر این فازها می پردازیم.

۱-۱- فاز inception

هدف این فاز شناخت مسئله است. در این فاز که نهایت چند هفته طول می کشد و کارها در ۱ یا ۲ تکرار انجام می شوند، با technical prototyping ، proof of concept prototyping و شناسایی ریسک ها، امکان سنجی[۱۱] انجام می‌شود. توجیه اقتصادی بررسی می شود. و نیازمندی های اساسی که حوزه سیستم را مشخص می کنند، شناسایی می‌شوند.

در این فاز مقصود و اهداف پروژه، اینکه سیستم باید ساخته یا خریداری شود، از روی سیستم های موجود ساخته شود یا از صفر و تخمین هزینه ها و ریسک ها انجام می شود. در نتیجه در انتهای این فاز تصمیم ادامه دادن یا ندادن سیستم گرفته می شود. همچنین در inception مدیریت و برنامه ریزی پروژه نیز انجام می شود.

---

[۱] Life cycle
[۲] workflow
[۳] requirements
[۴] analysis
[۵] design
[۶] implementation
[۷] test
[۸] scope
[۹] Business case
[۱۰] Baseline plan
[۱۱] feasibility

محصولاتی که در فاز inception تولید می شوند:

- ✓ سند چشم انداز[۱۲]
- ✓ مدل usecase اولیه
- ✓ واژه نامه پروژه
- ✓ طرح پروژه[۱۳] اولیه
- ✓ طرح توجیه اقتصادی[۱۴]
- ✓ سند ارزیابی ریسک[۱۵]
- ✓ معماری اولیه

۱-۲- فاز elaboration

هدف این فاز پیمایش قلمرو جواب است. در این فاز معماری مبنا قابل اجرا[۱۶] ساخته می شود؛ ارزیابی ریسک که در فاز قبلی ساخته شده بود، اصلاح می شود؛ معیار های کیفی مثل نرخ کشف نقص[۱۷] و تراکم نقص قابل قبول[۱۸]، تعریف می شوند؛ use case ها بطوری که ۸۰ درصد نیازمندی ها را پوشش دهند، ساخته می شوند. یک طرح[۱۹] تفصیلی برای فاز construction ایجاد می شود. پروپوزالی شامل منابع، زمان، تجهیزات، کارکنان و هزینه تهیه می شود.

تمرکز روی جریان های کاری هر تکرار در elaboration اینگونه است که در جریان کاری نیازمندی ها، حوزه سیستم و نیازمندی ها مشخص می شود؛ در تحلیل، ذات سیستم هدف مشخص می شود؛ در طراحی، معماری پایدار ساخته می شود؛ در اجرا، معماری مبنا پیاده سازی می شود؛ و در تست این معماری تست می شود.

محصولاتی که در فاز elaboration تولید می شوند:

- ✓ معماری مبنا قابل اجرا
- ✓ مدل ایستا UML
- ✓ مدل پویا UML

---

[۱۲] Vision document
[۱۳] Project plan
[۱۴] Business case
[۱۵] Risk assessment document
[۱۶] Executable architectural baseline
[۱۷] Defect discovery rate
[۱۸] Acceptable defect densities
[۱۹] plan

- ✓ مدل UML usecase
- ✓ سند چشم انداز
- ✓ سند ارزیابی ریسک بروزرسانی شده
- ✓ طرح توجیه اقتصادی بروزرسانی شده
- ✓ طرح برنامه بروزرسانی شده
- ✓ سند توافق

۱-۳- فاز construction

هدف در این فاز ارتقاء و تکامل مکرر محصولات ایجاد شده در فازهای قبلی است. در این فاز معماری مبنا ایجاد شده در فاز قبل به سیستم نهایی تبدیل می شود.

تمرکز روی جریان های کاری هر تکرار در construction اینگونه است که در جریان کاری نیازمندی ها، نیازمندی های باقیمانده شناسایی می شوند؛ در تحلیل، مدل تحلیل کامل می شود؛ در طراحی، مدل طراحی کامل می شود ؛ در اجرا، نسخه هدف پیاده سازی می شود؛ و در تست نسخه هدف تست می شود.

محصولاتی که در فاز construction تولید می شوند:

- ✓ محصول نرم افزاری
- ✓ مدل UML
- ✓ Test suite
- ✓ User manuals
- ✓ توضیحات نسخه ساخته شده
- ✓ طرح پروژه

۱-۴- فاز transition

هدف این فاز استقرار نهایی نرم افزار تولید شده در فاز قبلی است. در این فاز تست آلفا[۲۰]، تست بتا[۲۱]، تست سیستم[۲۲] و تست پذیرش[۲۳] در محیط کاربر انجام می شوند و مشکلات برطرف می شوند. بستر سخت افزاری و

[۲۰] Alpha testing
[۲۱] Beta testing
[۲۲] System testing
[۲۳] Acceptance testing

نرم افزاری محیط کاربر برای نرم افزار جدید آماده می شوند. در نرم افزار تغییراتی برای اینکه در محیط کاربر کار کند، داده می شود. تیم برای مشاوره به کاربر آماده شود. بررسی پس از پروژه[۲۴] مثل post-mortem انجام می‌شود.

تمرکز روی جریان های کاری هر تکرار در transition اینگونه است که در جریان کاری نیازمندی ها، کاری انجام نمی شود؛ در تحلیل، در صورت نیاز، مدل تحلیل بروزرسانی می شود؛ در طراحی، در صورت بروز مشکل در هنگام تست، مدل طراحی اصلاح می شود؛ در اجرا، نرم افزار برای اجرا در محیط کاربر آماده می شود و مشکلات کشف شده در تست، برطرف می شوند؛ و در تست، تست آلفا، تست بتا، تست سیستم و تست پذیرش در محیط کاربر انجام می شود.

محصولاتی که در فاز transition تولید می شوند:

- ✓ محصول نرم افزاری
- ✓ طرح ساپورت[۲۵] کاربر
- ✓ User manual های بروزرسانی شده

[۲۴] Post-project review
[۲۵] Support plan

۲- متدولوژی Booch

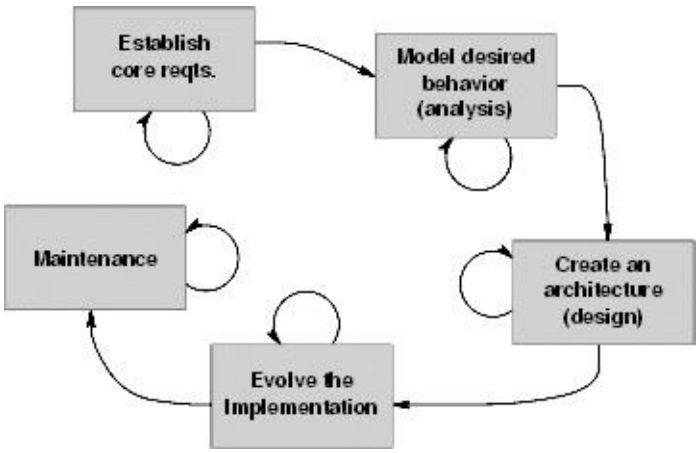
متدولوژی Booch در دو نسخه در سال های ۱۹۹۱ و ۱۹۹۴ توسط Booch معرفی شد. در نسخه اول تنها به عنوان یک روش طراحی در قلمرو جواب بود در حالی که در نسخه بعدی تحلیل نیز اضافه شد. این متدولوژی اولین متدولوژی تکراری-افزایشی[۲۶] است.

متدولوژی Booch طراحی شیء گرا را به عنوان یک فرایند تکرار شونده در یک فرایند تکراری در سطح چرخه عمر مدلسازی کرده است. که این دو به عنوان micro process و macro process معرفی شدند. macro process به عنوان یک چارچوب کنترلی برای micro process تعریف شده است، یعنی با استفاده از macro process ، micro process را مدیریت می کنیم. در ادامه به توضیح جزئی تر این دو می پردازیم.

۲-۱- macro process

همانطور که ذکر شد macro process یک چارچوب کنترلی و مدیریتی برای micro process می باشد، بنابراین بیشتر بخش هایش فعالیت های مدیریت نرم‌افزار اساسی مانند تضمین کیفیت، مرور کد و مستندات است. همچنین macro process بیشتر بر مشتریان و خواسته های آن ها برای مواردی مانند کیفیت، کامل بودن و زمان‌بندی، تمرکز می کند. فعالیت های تیم توسعه در این بخش در مقیاس هفته ها تا ماه ها است.

macro process را در شکل ۱ مشاهده می کنید.



شکل ۱- macroprocess

[۲۶] iterative-incremental

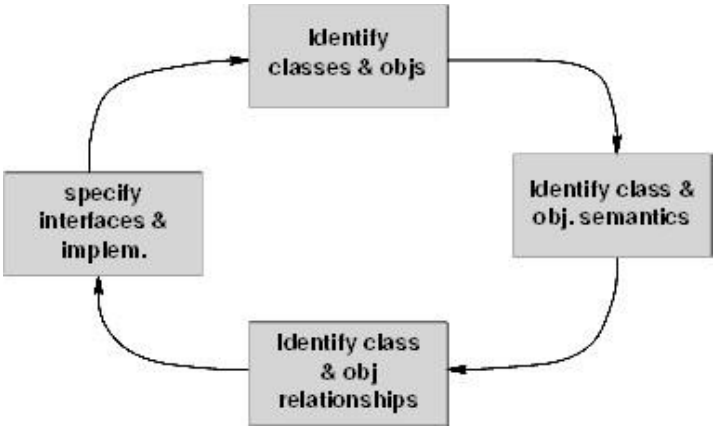
همانطور که در شکل ۱ مشاهده می کنید، macro process از ۵ فاز تشکیل شده است. در هر فاز micro process بصورت تکراری انجام می شود. فاز ها عبارتند از:

۱. مفهوم سازی: نیازمندی های اصلی نرم افزار استخراج می شوند.
۲. تحلیل: یک مدل از رفتار مطلوب سیستم ایجاد می شود.
۳. طراحی: یک معماری برای پیاده سازی ایجاد می شود.
۴. تکامل: پیاده سازی تکامل پیدا می کند.
۵. نگهداری: تکامل پس از تحویل[۲۷] مدیریت می شود.

micro process -۲-۲

در micro process فعالیت های ریزدانه ایجاد نرم افزار به صورت روزانه توسط فرد یا گروه کوچکی از توسعه‌دهندگان انجام می شوند. micro process بر اساس سناریوها و مشخصات معماری که از macro process پدید می آیند، انجام می شود.

micro process را در شکل ۲ مشاهده می کنید.



شکل ۲- micro process

همانطور که در شکل ۲ مشاهده می کنید، micro process از ۴ مرحله تشکیل شده است که بصورت تکراری در یک حلقه انجام می شوند.

[۲۷] post-delivery

این مراحل عبارتند از:

۱. شناسایی کلاس ها و شیء ها:

کلاس ها و اشیاء از طریق تعیین مرزهای مسئله، محدود کردن مسئله و یافتن انتزاعات[۲۸] در قلمرو مسئله، شناسایی می شوند. کلاس‌هایی که در مراحل اولیه macro process شناسایی می‌شوند، عمدتاً به قلمرو مسئله تعلق دارند، در حالی که کلاس‌هایی که در طول طراحی اضافه می‌شوند، معمولاً به قلمرو پیاده‌سازی متعلق هستند.

در این مرحله دیکشنری داده[۲۹] تولید می شود که همه کلاس ها و اشیاء در طول توسعه سیستم را نگه می‌دارد. همچنین دیکشنری داده ممکن است در طول توسعه تغییر کند.

۲. شناسایی معناشناسی[۳۰] کلاس ها و شیء ها:

برای ایجاد معانی کلاس ها و اشیاء شناسایی شده در مرحله قبل، جنبه رفتاری آن ها بررسی می شود. برای اینکار رفتار و ویژگی[۳۱] های هر انتزاعی که در مرحله قبل شناسایی شده است، ایجاد می شوند و انتزاعات اصلاح می‌شوند. همچنین مسئولیت[۳۲] ها به انتزاع ها اضافه می شوند و عملیات[۳۳] ذکر شده برای هر کلاس ایجاد می شود.

در این مرحله نمودار های حالت[۳۴]، نمودار های شیء[۳۵] و نمودار های تعامل[۳۶] ساخته می شوند. نمودارهای حالت برای کلاس هایی که رفتار پویا دارند، تولید می شود. نمودار های شیء و تعامل در واقع همریخت هستند و الگوهای تعامل بین اشیاء را به تصویر می کشند. با این تفاوت که نمودار شیء بر روابط ایستا بین اشیاء تاکید دارد، در حالی که نمودار تعامل بر توالی تعاملات بین اشیاء تاکید دارد. همچنین نمودارهای شیء در مراحل اولیه macro process برای نشان دادن روابط ساختاری بین اشیاء نیز مفید هستند.

---

[۲۸] abstractions
[۲۹] data dictionary
[۳۰] semantic
[۳۱] attribute
[۳۲] responsibility
[۳۳] operation
[۳۴] state diagrams
[۳۵] object diagrams
[۳۶] interaction diagrams

۳. شناسایی روابط کلاس ها و شیء ها:

برای نشان دادن روابط بین کلاس ها و اشیاء، نحوه تعامل اشیا در سیستم به طور دقیق مشخص می‌شود. الگو[۳۷]های موجود درکلاس ها که اجازه سازماندهی مجدد و ساده سازی ساختار کلاس را می‌دهند، جستجو می شوند و تصمیمات دید[۳۸] گرفته می شود.

در این مرحله نمودارهای کلاس[۳۹]، نمودار های ماژول[۴۰] و نمودار های شیء ایجاد می شوند. نمودارهای کلاس، کلاس ها و روابط آن ها را نشان می دهند و نمودارهای ماژول، ماژول های فیزیکی و وابستگی‌های بین آن ها را نشان می دهند و معماری فیزیکی سیستم را به تصویر می کشند. این نمودار معمولاً در مراحل بعدی macro process ساخته می شود.

۴. شناسایی رابط[۴۱] و پیاده سازی کلاس ها و شیء ها:

برای نمایش کلاس ها و اشیائی که تاکنون شناسایی شده اند، تصمیمات طراحی گرفته می شود. کلاس‌ها و اشیاء به ماژول ها و فرایندهای پیاده سازی این ماژول ها به پردازنده ها تخصیص داده می‌شوند. همچنین به نمودارهای ماژول جزئیات اضافه می شود.

در این مرحله نمودار های فرایند[۴۲] تولید می شوند. این نمودار با به تصویر کشیدن پردازنده ها، دستگاه‌ها و اتصالات آن ها، معماری پلتفرم سخت افزاری سیستم را نشان می دهد. همچنین نشان می‌دهد که کدام فرایندها به هر پردازنده اختصاص داده شده است.

---

[۳۷] pattern
[۳۸] visibility
[۳۹] class diagrams
[۴۰] module diagrams
[۴۱] interface
[۴۲] process diagram

۳- متدولوژی OMT (Object Modeling Technique)

OMT در سال ۱۹۹۱ توسط Rumbaugh و همکارانش معرفی شد. این متدولوژی از سه روش مدلسازی ساختاری، رفتاری و وظیفه ای استفاده می کند و سپس روشی برای ترکیب آن ها ارائه می کند. از همین جهت به این متدولوژی ترکیبی[۴۳] می گویند.

سه مدلی که به آن ها اشاره شد عبارتند از:

- Object Model (OM) : ساختار ایستا[۴۴] سیستم را نشان می دهد. کلاس های اشیاء و ارتباط بینشان را توسط Class Diagram نمایش می دهد.
- Dynamic Model (DM) : رفتار سیستم را با بررسی رفتار اشیاء در طول زمان و جریان کنترل و رویدادها در بین اشیاء توسط Event-Trace Diagrams و State Transition Diagrams به تصویر می کشد. Event-Trace Diagram تراکنش های رفتاری را به شکل جریان های رویداد ها نشان می دهد و State Transition Diagram حالت هایی که سیستم در آن ها قرار می گیرد و تغییر حالت ها در اثر آمدن event های بیرونی نشان می دهد؛ همچنین گویاء اینکه هر کلاس از زمان تشکیل شئ هایش چه حالت‌هایی را می گذراند نیز می باشد.
- Functional Model (FM) : فرایندهای داخلی سیستم را بدون توضیحات در مورد نحوه انجام این فرایندها در مجموعه ای سلسله مراتبی از نمودارهای جریان داده (DFD) توصیف می کند. DFD واحدهای کاری و داده هایی که بین آن ها رد و بدل می شود را نشان می دهد.

فرایند OMT ۵ فاز دارد:

۱. Analysis
۲. System Design
۳. Object Design
۴. Coding
۵. Testing

۳-۱- تحلیل[۴۵]

در این مرحله هدف این است که یک مدل صحیح از دنیای واقع بسازیم. این کار در ۴ مرحله انجام می شود:

- Object Model را می سازیم.

---

[۴۳] combinative
[۴۴] static
[۴۵] analysis

- Dynamic Model را می سازیم.
- Functional Model را می سازیم.
- این سه مدل را صحت سنجی، تکرار و اصلاح می کنیم.

۳-۲- طراحی سیستم[۴۶]

در این فاز ساختار سطح بالای سیستم و ویژگی های قلمرو جواب مشخص می شود. به این صورت که:

سیستم ها به زیر سیستم ها سازماندهی می شوند، نحوه ی برقراری هم روندی مشخص می شود، زیر سیستم‌ها به پردازنده ها تخصیص پیدا می کنند، برای پیاده سازی ذخیره سازی داده ها استراتژی مشخص می شود، منابع سراسری و نحوه ی دسترسی به آن ها مشخص می شود، نحوه ی اجرای کنترل نرم افزار، برای مثال برای مدیریت تعامل زیرسیستم ها با یکدیگر و مدیریت دستیابی آن ها به منابع سراسری، مشخص می شود، شرایط مرزی (start up – shutdown – exception) در نظر گرفته می شود، اولویت های trade-off تعیین می شود.

۳-۳- طراحی شیء[۴۷]

در این مرحله ۳ مدل ذکر شده کامل می شوند و کلاس های قلمرو جواب و روابط بینشان به آن ها اضافه می‌شود. در واقع جزئیات برای تعیین نحوه پیاده سازی مشخص می شود.

[۴۶] System Design
[۴۷] Object Design

۴- مقایسه متدولوژی USDP (UP) با دو متدولوژی Booch و OMT

۴-۱- معیار های فرایندی

۴-۱-۱- تعریف

این معیار بیان می کند که متدولوژی باید به خوبی مستند شده باشد و دارای توضیحات جامع، واضح، منطقی، صحیح، با جزئیات و سازگار باشد. به همین جهت باید چیستی و چگونگی کار مطرح شوند.

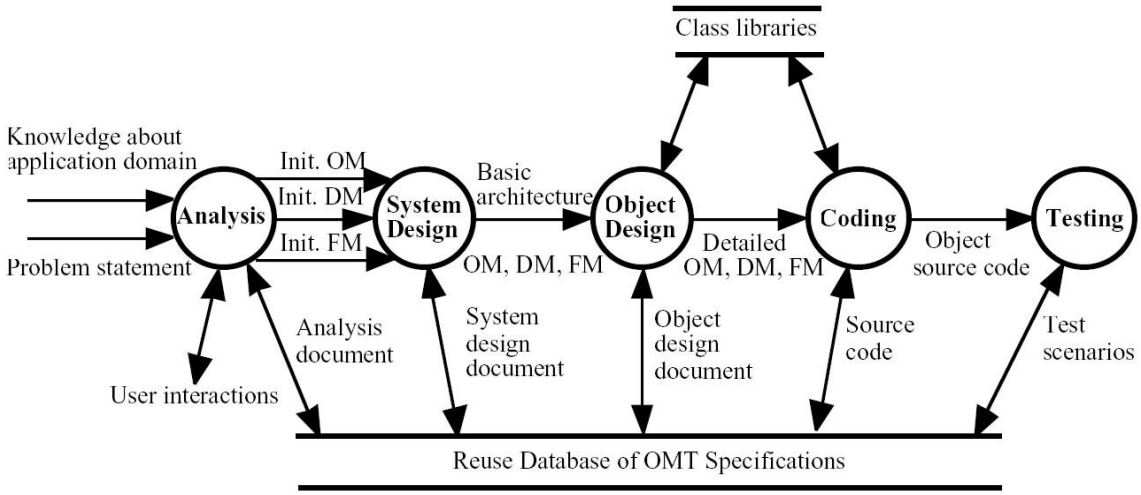
- چرخه عمر و واحد های کاری:

متدولوژی UP : چرخه عمر این متدولوژی حاوی ۴ فاز inception ، elaboration ، construction و transition می باشد که در هرکدام از این فاز ها iteration هایی انجام می شوند. هر iteration شامل ۵ جریان کاری است که عبارتند از نیازمندی ها، تحلیل، طراحی، اجرا و تست. در فاز inception چشم انداز، حوزه و توجیه اقتصادی پروژه تعریف می شود. در فاز elaboration تعریف محصول، اصلاح می شود؛ معماری مبنا تعریف می شود؛ و یک برنامه دقیق تر برای توسعه و استقرار ایجاد می شود. در فاز construction محصول تا حدی که قابل تحویل به کاربر باشد ساخته می شود. و در فاز transition انتقال محصول به محیط کاربر صورت می گیرد که شامل ساخت، تحویل، آموزش و برنامه ریزی برای پشتیبانی و نگهداری محصول می‌باشد. جزئیات هر فاز بطور کامل در متدولوژی آمده است. بنابراین متدولوژی UP چرخه عمر و واحدهای کاری خود را با جزئیات بیان کرده است.

متدولوژی Booch : این متدولوژی یک متدولوژی تکراری-افزایشی است که در آن فرایند را به عنوان یک فرایند تکرار شونده (micro process) در یک فرایند تکراری در سطح چرخه عمر (macro process) مدلسازی کرده است. macro process به عنوان یک چارچوب کنترلی برای micro process تعریف شده است. macro process از ۵ فاز تشکیل شده است. در هر فاز micro process بصورت تکراری انجام می شود. ۵ فاز آن عبارتند از مفهوم‌سازی، تحلیل، طراحی، تکامل و نگهداری (جزئیات در بخش ۱-۲). خود micro process از ۴ مرحله تشکیل شده است که بصورت تکراری انجام می شوند. این مرحله ها عبارتند از شناسایی کلاس ها و شیء ها، شناسایی معناشناسی کلاس ها و شیء ها، شناسایی روابط کلاس ها و شیء ها و شناسایی رابط[۴۸] و پیاده‌سازی کلاس ها و شیء‌ها (جزئیات در بخش ۲-۲). بنابراین در Booch واحد های کاری و چرخه عمر آن توضیح داده شده است.

[۴۸] interface

متدولوژی OMT : در این متدولوژی چرخه ی عمر و واحد های کاری کاملا توضیح داده شده اند. به این صورت که OMT شامل ۵ فاز تحلیل، طراحی سیستم، طراحی شیء، کد و تست می باشد. در فاز تحلیل هدف این است که یک مدل صحیح از دنیای واقع بسازیم، در همین جهت سه مدل شیء، پویا و وظیفه ای به طور تکراری ساخته می‌شوند تا جنبه های ساختاری، رفتاری و وظیفه ای سیستم را نشان دهند. در فاز طراحی سیستم یا همان طراحی معماری، ساختار سطح بالای سیستم و ویژگی های قلمرو جواب مشخص می شود. در فاز طراحی شیء، کلاس‌های قلمرو جواب و روابط بینشان به سه مدلی که ساختیم اضافه می‌شود. در واقع در این فاز طراحی تفضیلی انجام می شود. در فاز کد و تست هم که پیاده سازی و آزمون نرم افزار انجام می‌شود. جزئیات بیشتر در مورد کار ها در فاز تحلیل، طراحی سیستم و طراحی شیء نیز بیان شده است (بخش ۱). یک نموداری هم برای فرایند آن آورده شده است که در شکل ۳ مشاهده می کنید.



شکل ۳- متدولوژی OMT

مقایسه متدولوژی UP و Booch :

شباهت ها:

هر دو متدولوژی فرایند تکراری-افزایشی دارند. هر دو فاز های تحلیل، طراحی، پیاده سازی، تست و استقرار را با نام های مختلف دارند. جزئیات کارهایی که باید در هر فاز و فعالیت های تکراری انجام شود در هر دو متدولوژی آمده است.

تفاوت ها:

متدولوژی Booch دارای فاز نگهداری است در حالی که متدولوژی UP فازی جداگانه برای آن در نظر نگرفته است. فعالیت های تکراری که در فازهای این دو متدولوژی تعریف شده اند با هم متفاوتند.

مقایسه:

می توان گفت که متدولوژی Booch به دلیل اینکه فاز نگهداری تعریف کرده است نسبت به متدولوژی UP در این قسمت برتری دارد. گرچه متدولوژی Booch نیز فعالیت های جداگانه برای فاز نگهداری در نظر نگرفته است. هر دو متدولوژی جزئیات واحدهای کاری خود را بیان کردند اما متدولوژی UP از جزئیات بیشتری برخوردار است و برای مثال در هر فاز شرایط تحقق و محصولات را به خوبی تعریف می کند. بنابراین می توان گفت که از نظر جزئیات از متدولوژی Booch برتری دارد.

مقایسه متدولوژی UP و OMT :

شباهت ها:

هر دو متدولوژی فاز های تحلیل، طراحی، پیاده سازی و تست را با نام های مختلف دارند. جزئیات کارهایی که باید در هر فاز و فعالیت های تکراری انجام شود در هر دو متدولوژی آمده است.

تفاوت ها:

فرایند متدولوژی UP تکراری-افزایشی است در حالی که در متدولوژی OMT اینطور نیست. متدولوژی UP دارای فاز استقرار است در حالی که متدولوژی OMT فازی برای آن در نظر نگرفته است.

مقایسه:

متدولوژی UP با جزئیات بیشتری واحدهای کاری خود را بیان کرده است. فرایند آن تکراری-افزایشی است و همچنین دارای فاز استقرار است که در متدولوژی OMT آن را مشاهده نمی کنیم. بنابراین از این نظر ها نسبت به متدولوژی OMT برتری دارد.

- تولیدکنندگان:

متدولوژی UP : این متدولوژی در یک حالت متدولوژی را بصورت role-centered بیان کرده است و نقش افراد در کارها مشخص شده است.

متدولوژی Booch : در این متدولوژی اشاره ای به افراد و نقششان در فعالیت ها نشده است.

متدولوژی OMT : در این متدولوژی نیز اشاره ای به افراد و نقششان در فعالیت ها نشده است.

مقایسه متدولوژی UP و Booch :

شباهت ها: –

تفاوت ها:

متدولوژی UP افراد و نقش هایشان را مشخص کرده است در حالی که متدولوژی Booch اشاره‌ای به افراد و نقششان نکرده است.

مقایسه:

متدولوژی UP در تعریف تولیدکنندگان نسبت به متدولوژی Booch برتری دارد.

مقایسه متدولوژی UP و OMT :

شباهت ها: –

تفاوت ها:

متدولوژی UP افراد و نقش هایشان را مشخص کرده است در حالی که متدولوژی OMT اشاره‌ای به افراد و نقششان نکرده است.

مقایسه:

متدولوژی UP در تعریف تولیدکنندگان نسبت به متدولوژی OMT برتری دارد.

- زبان مدلسازی:

متدولوژی UP : این متدولوژی بر مبنای زبان مدلسازی UML ساخته شده است.

متدولوژی Booch : این متدولوژی از زبان مدلسازی مخصوص به خود برای مدلسازی و ایجاد محصولات استفاده کرده است که توضیحات آن در متدولوژی بیان شده است.

متدولوژی OMT : این متدولوژی سه مدل شیء، پویا و وظیفه ای برای نشان دادن جنبه های ساختاری، رفتاری و وظیفه‌ای سیستم، می سازد. در مدل شیء کلاس های اشیاء و ارتباط بینشان را توسط Class Diagram نمایش می دهد؛ در مدل پویا رفتار سیستم را توسط Event-Trace Diagrams و State Transition Diagrams به تصویر می کشد. و در مدل وظیفه ای، وظیفه مندی سیستم را با نمودار DFD (که شیء گرا نیست) نشان می دهند.

مقایسه متدولوژی UP و Booch :

شباهت ها: –

تفاوت ها:

متدولوژی UP از زبان مدلسازی UML استفاده کرده است در حالی که متدولوژی Booch زبان مدلسازی مخصوص به خود را تعریف کرده است.

مقایسه:

زبان مدلسازی متدولوژی UP زبان غنی UML است که از این لحاظ نسبت به متدولوژی Booch برتری دارد.

مقایسه متدولوژی UP و OMT :

شباهت ها: –

تفاوت ها:

متدولوژی UP از زبان مدلسازی UML استفاده کرده است در حالی که متدولوژی OMT دارای ۳ مدل شیء، پویا و وظیفه ای است که دیدگاه ساختاری، رفتاری و وظیفه ای سیستم را نشان می دهند. همچنین در مدل وظیفه ای خود از DFD استفاده کرده است که شیء گرا نیست.

مقایسه:

زبان مدلسازی متدولوژی UP زبان غنی UML است که از این لحاظ نسبت به متدولوژی Booch برتری دارد. اما در UML مدلسازی وظیفه ای تنها با use case انجام می شود که گاهی اوقات کافی نیست. بنابراین وجود DFD در متدولوژی OMT می تواند در بعضی موارد برتری آن نسبت به متدولوژی UP باشد.

- محصولات:

متدولوژی UP : در این متدولوژی محصولات متعددی در طی فازها ایجاد می شوند. در فاز inception، سند چشم‌انداز، مدل use case اولیه، واژه نامه پروژه، project plan اولیه، طرح توجیه اقتصادی[۴۹]، سند ارزیابی ریسک[۵۰] و معماری اولیه، تولید می شوند. در فاز elaboration، معماری مبنا قابل اجرا، مدل ایستا UML، مدل پویا UML، مدل UML use case، سند چشم انداز، سند ارزیابی ریسک بروزرسانی شده، طرح توجیه اقتصادی بروزرسانی شده، طرح برنامه بروزرسانی شده و سند توافق، تولید می شوند. در فاز construction، محصول نرم افزاری، مدل UML، test suite، user manual ها، توضیحات نسخه ساخته شده و project plan ، تولید می شوند. و در فاز transition، محصول نرم افزاری، برنامه ساپورت[۵۱] کاربر و user manual های بروزرسانی شده، تولید می شوند.

متدولوژی Booch : در این متدولوژی در micro process که فعالیت های ریزدانه ایجاد نرم افزار انجام می‌شوند محصولاتی تولید می شوند. در مرحله اول که کلاس ها و اشیاء شناسایی می شوند، دیکشنری داده تولید می شود که همه کلاس ها و اشیاء در طول توسعه سیستم را نگه می‌دارد. در مرحله دوم که معناشناسی[۵۲] کلاس ها و اشیاء مشخص می شود و در واقع مدلسازی رفتاری صورت می گیرد، نمودار های حالت[۵۳] ،برای کلاس هایی که رفتار پویا دارند، و نمودار های شیء[۵۴] و نمودار های تعامل[۵۵] ، برای نشان دادن الگوهای تعامل بین اشیاء، ساخته می شوند. نمودار های شیء و تعامل در واقع همریخت هستند با این تفاوت که نمودار شیء بر روابط ایستا بین اشیاء تاکید دارد، در حالی که نمودار تعامل بر توالی تعاملات بین اشیاء تاکید دارد. در مرحله سوم که روابط کلاس ها و اشیاء شناسایی می‌شوند و در واقع مدلسازی ساختاری سیستم انجام می‌شود،

[۴۹] Business case
[۵۰] Risk assessment document
[۵۱] Support plan
[۵۲] semantic
[۵۳] state diagrams
[۵۴] object diagrams
[۵۵] interaction diagrams

نمودارهای کلاس[۵۶]، برای نشان دادن کلاس ها و روابط بین آن ها، نمودار های ماژول[۵۷] ، برای نشان دادن ماژول های فیزیکی و وابستگی‌های بین آن ها و نمودار های شیء، ایجاد می شوند. و در مرحله آخر که رابط[۵۸]ها شناسایی می شوند و کلاس ها و اشیاء پیاده سازی می شوند، نمودار های فرایند[۵۹] ، برای به تصویر کشیدن پردازنده‌ها، دستگاه‌ها و اتصالات آن ها، تولید می شوند.

متدولوژی OMT : در این متدولوژی محصولاتی که در هر گام تولید می شوند مشخص شده اند. همانطور که در شکل ۳ می بینید خروجی هر فاز مشخص است. در فاز تحلیل مدل های شیء، پویا و وظیفه ای اولیه ساخته می شوند که در هر کدام محصولاتی که تولید می شوند class diagram و data dictionary در مدل شیء، Event-Trace Diagrams و State Transition Diagrams در مدل پویا و نمودارهای DFD در مدل وظیفه ای می‌باشند. در فاز طراحی یک معماری در سطح سیستم ساخته می شود و ۳ مدل ذکر شده به همراه نمودارهایشان بروزرسانی می شوند. در فاز طراحی شیء مدل های ذکر شده به همراه نمودار هایشان کامل می شوند و کلاس های قلمرو جواب به آن ها اضافه می شود. در فاز برنامه نویسی سورس کد تولید می شود و در نهایت در فاز تست سناریو های آزمون تولید می شود.

مقایسه متدولوژی UP و Booch :

شباهت ها:

در هر دو متدولوژی محصولات به خوبی تعریف شده اند. محصولاتی مثل نمودار کلاس، نمودار شیء، نمودار ماژول و نمودار حالت از نظر مفهوم شبیه نمودار های کلاس، شیء، پکیج و انتقال حالت[۶۰] UML هستند.

تفاوت ها:

به جز محصولات بالا سایر محصولات با یکدیگر متفاوت هستند. همچنین در متدولوژی UP مستنداتی مثل سند چشم انداز، توجیه اقتصادی، ارزیابی ریسک و ... تولید می شوند که در متدولوژی Booch خبری از آن‌ها نیست.

[۵۶] class diagrams
[۵۷] module diagrams
[۵۸] interface
[۵۹] process diagram
[۶۰] State transition diagram

مقایسه:

متدولوژی UP به دلیل استفاده از زبان مدلسازی UML محصولات بهتری تولید می کند و همچنین در کنار آن ها مستنداتی نیز ایجاد می کند. بنابراین متدولوژی UP از نظر کیفیت محصولات و کامل بودن آن ها از متدولوژی Booch بهتر عمل می کند.

مقایسه متدولوژی UP و OMT :

شباهت ها:

محصولات class diagram و state transition diagram در هر دو متدولوژی دیده می شود.

تفاوت ها:

سایر محصولات این دو متدولوژی با هم متفاوتند. همچنین در متدولوژی UP مستنداتی نیز به عنوان محصول تولید می شوند که در متدولوژی OMT دیده نمی شود.

مقایسه:

متدولوژی UP به دلیل استفاده از زبان مدلسازی UML محصولات بهتر و جامع تری تولید می کند و همچنین در کنار آن ها مستنداتی نیز ایجاد می کند. بنابراین متدولوژی UP از نظر کامل بودن محصولات از متدولوژی OMT بهتر عمل می کند.

- تکنیک ها و قوانین: تکنیک درواقع به چگونگی انجام کار ها اشاره دارد.

متدولوژی UP : همانطور که ذکر شد این متدولوژی از ۴ فاز تشکیل شده که در هر کدام از این فاز ها ۵ جریان کاری با تمرکز های مختلف متناسب با فازی که در آن قرار داریم، بصورت تکراری اجرا می‌شوند. و در هر فاز برای تحقق اهداف آن فاز به تکنیک ها و قوانین انجام کارها اشاره شده است.

متدولوژی Booch : همانطور که ذکر شد، متدولوژی Booch یک متدولوژی تکراری-افزایشی است که در آن macro process به عنوان یک چارچوب کنترلی برای micro process تعریف شده است. Macro process بیشتر بخش هایش فعالیت های مدیریت نرم‌افزار اساسی مانند تضمین کیفیت، مرور کد و مستندات است که در مقیاس هفته ها و ماه ها انجام می شود در حالی که در micro process فعالیت های ریزدانه ایجاد نرم افزار، بر اساس سناریوها و مشخصات معماری که از macro process پدید می آیند، به صورت روزانه توسط

فرد یا گروه کوچکی از توسعه‌دهندگان انجام می شوند. Macro process و micro process مراحلی دارند که چگونگی انجام آن ها در متدولوژی ذکر شده است. بنابراین Booch تکنیک ها و قوانین را مشخص کرده است.

متدولوژی OMT : این متدولوژی همانطور که ذکر شد، حاوی ۵ فاز تحلیل، طراحی سیستم، طراحی شیء، کد و تست است و چگونگی انجام مراحل تحلیل، طراحی سیستم و طراحی شیء توضیح داده شده است (به بخش ۳ مراجعه کنید).

مقایسه متدولوژی UP و Booch :

شباهت ها:

هر دو متدولوژی تکنیک ها و قوانین مربوط به خود را تعریف کردند.

تفاوت ها:

تکنیک ها و قوانین در این دو متدولوژی متفاوت است.

مقایسه:

هر دو متدولوژی متناسب با فاز ها و فعالیت هایشان تکنیک ها و قوانین خاص خود را تعریف کرده اند بنابراین از این نظر نسبت به یکدیگر برتری ندارند.

مقایسه متدولوژی UP و OMT :

شباهت ها:

هر دو متدولوژی تکنیک ها و قوانین مربوط به خود را تعریف کردند.

تفاوت ها:

تکنیک ها و قوانین در این دو متدولوژی متفاوت است.

مقایسه:

هر دو متدولوژی متناسب با فاز ها تکنیک ها و قوانین خاص خود را تعریف کرده اند بنابراین از این نظر نسبت به یکدیگر برتری ندارند.

- دغدغه های مربوط به فعالیت های چتری:

فعالیت های چتری به طور کامل در بخش ۳-۱-۴ توضیح داده شده اند، اما به طور مختصر به فعالیت هایی گفته می شود که در کل طول چرخه عمر نرم افزار برای مدیریت ریسک، مدیریت پروژه و تضمین کیفیت صورت می گیرند.

متدولوژی UP : این متدولوژی فعالیت های چتری که برای مدیریت ریسک، مدیریت پروژه و تضمین کیفیت انجام می شوند را پوشش می دهد.

- برای مدیریت ریسک، فعالیت هایی مثل تحلیل امکان سنجی، برنامه ریزی مبتنی بر ریسک، ایجاد prototype های اولیه، مشارکت فعال کاربر،V&V مستمر و یکپارچه سازی مداوم انجام می شوند. همچنین فرایند این متدولوژی تکراری-افزایشی است که به مدیریت ریسک کمک می‌کند.
- برای مدیریت پروژه در فاز inception فعالیت‌های برنامه ریزی و مدیریت پروژه شامل نمودارهای گانت، برنامه ها، بودجه ها و غیره، انجام می شوند. در انتهای فاز elaboration برنامه دقیق برای iteration های اولیه فاز بعدی یعنی فاز construction، ایجاد می‌شود و برنامه ریزی مابقی iterationها در طول این فاز کامل می شود. بنابراین این برنامه ریزی ها بطور مکرر در طی فازها مورد بازبینی و تجدید نظر قرار می گیرند. البته این متدولوژی به مدیریت افراد اشاره ای نکرده است اما مدیریت پروژه را به خوبی انجام می دهد.
- برای تضمین کیفیت، V&V مستمر و ردیابی به نیازمندی ها صورت می گیرد. همچنین در فاز elaboration معیار های کیفی مثل نرخ کشف نقص و تراکم نقص قابل قبول، تعریف می شوند که کیفیت کد را نشان می دهند و با پایین نگه داشتن این معیار ها می توان کیفیت را تضمین کرد.

متدولوژی Booch : این متدولوژی اشاره ای به کارهایی که در جهت مدیریت ریسک انجام می شوند، نکرده است. اما فرایند این متدولوژی، یک فرایند تکراری-افزایشی است. در هر فاز macro process ، که یک چارچوب مدیریتی برای macro process است، micro process بصورت تکراری انجام می شود و این مدیریت ریسک را در صورت بروز آن ساده تر می کند.

در این متدولوژی فعالیت های مدیریتی در macro process انجام می شوند. برای مثال فعالیت هایی مثل تضمین کیفیت، مرور کد و مستندات در این بخش انجام می شوند. همچنین در macro process ، برنامه‌ریزی، زمانبندی و کنترل نیز انجام می شود. اما Booch اشاره ای به مدیریت اعضای تیم نکرده است. اما در کل مدیریت پروژه در این متدولوژی در سطح خوبی قرار دارد.

در متدولوژی Booch همانطور که ذکر شد، فعالیت هایی مثل تضمین کیفیت، مرور کد و مستندات در macro process انجام می شوند. بنابراین مرور کد بصورت تکراری انجام می شود و همچنین کیفیت مد نظر کاربر چک می شود. بنابراین این متدولوژی تضمین کیفیت دارد.

متدولوژی OMT : این متدولوژی اشاره ای به کارهایی که در جهت مدیریت ریسک انجام می شوند، نکرده است. اما در فاز تحلیل سه مدل شیء، پویا و وظیفه ای پس از ساخته شدن، صحت سنجی می شوند که می‌تواند کمی به مدیریت ریسک کمک کند.

در این متدولوژی برنامه ریزی، زمانبندی و کنترل انجام نمی شوند. اشاره ای هم به استفاده از تکنیک هایی مثل بازبینی مکرر برنامه ها و مدیریت تیم نشده است.

فعالیت ها در OMT حاوی ارزیابی کیفیت نمی باشند. از تکنیک هایی مثل مرور فنی تکراری[۶۱]، طراحی بر اساس قرارداد و تکنیک‌هایی که قابلیت ردیابی نیازمندی‌ها را افزایش می‌دهند، استفاده ای نشده است. تنها کاری که صورت می گیرد صحت سنجی مدل های شیء، پویا و وظیفه ای در فاز تحلیل است که می‌تواند کمی به تضمین کیفیت کمک کند.

مقایسه متدولوژی UP و Booch :

شباهت ها:

- ✓ هر دو متدولوژی فرایند تکراری-افزایشی دارند که به مدیریت ریسک کمک می کند. هر دو نیز در فعالیت های تکراریشان تست انجام می دهند که در آن V&V مستمر انجام می شود.
- ✓ هر دو متدولوژی فعالیت های برنامه ریزی، زمانبندی و کنترل را انجام می دهند بنابراین مدیریت پروژه صورت می گیرد اما در هیچ یک به مدیریت افراد درگیر پروژه اشاره نشده است.
- ✓ هر دو متدولوژی فعالیت هایی برای تضمین کیفیت انجام می دهند و سطح آن در هر دو بالاست.

تفاوت ها:

- ✓ متدولوژی UP از تکنیک های تحلیل امکان سنجی اولیه، نمونه سازی اولیه، برنامه‌ریزی مبتنی بر ریسک و مشارکت فعال کاربر که برای مدیریت ریسک مفید هستند، استفاده می کند. در حالی که متدولوژی Booch از هیچ کدام از آن ها استفاده نمی کند.

[۶۱] iterative technical reviews

- ✓ در متدولوژی UP، plan به طور مستمر مورد مرور و بازنگری قرار می گیرد در حالی که در متدولوژی Booch اشاره ای به آن نشده است.
- ✓ متدولوژی UP ردیابی به نیازمندی ها را بطور مستقیم دارد و همچنین معیار هایی نیز برای سنجش کیفیت تعریف کرده است. در حالی که متدولوژی Booch این ها را ندارد.

مقایسه:

متدولوژی UP کاملا مدیریت ریسک را انجام می دهد و از این جهت نسبت به متدولوژی Booch برتری دارد. هر دو متدولوژی مدیریت پروژه و تضمین کیفیت را انجام می دهند اما متدولوژی UP به دلیل تفاوت‌هایی که ذکر شد بهتر عمل می کند و نسبت به متدولوژی Booch برتری دارد.

مقایسه متدولوژی UP و OMT :

شباهت ها: –

تفاوت ها:

- ✓ متدولوژی UP از تکنیک های تحلیل امکان سنجی اولیه، نمونه سازی اولیه، برنامه‌ریزی مبتنی بر ریسک، مشارکت فعال کاربر، V&V مستمر و یکپارچه سازی مداوم که برای مدیریت ریسک مفید هستند، استفاده می‌کند. در حالی که متدولوژی OMT از هیچ یک از آن ها استفاده نمی کند.
- ✓ متدولوژی UP فعالیت های برنامه ریزی، زمانبندی و کنترل را انجام می دهند بنابراین مدیریت پروژه صورت می‌گیرد. همچنین در آن plan به طور مستمر مورد مرور و بازنگری قرار می گیرد.
- ✓ متدولوژی UP، V&V مستمر و ردیابی به نیازمندی ها را بطور مستقیم دارد و همچنین معیار هایی نیز برای سنجش کیفیت تعریف کرده است. در حالی که در متدولوژی OMT تنها کاری که انجام می شود صحت سنجی مدل های شیء، پویا و وظیفه ای و همچنین فاز تست است که می تواند کمی به تضمین کیفیت کمک کند.

مقایسه:

متدولوژی UP به خوبی از فعالیت های چتری یعنی مدیریت ریسک، مدیریت پروژه و تضمین کیفیت پشتیبانی می کند در حالی که متدولوژی OMT هیچ یک از آن ها را ندارد و این یکی از برتری های متدولوژی UP نسبت به آن است.

۴-۱-۲- پوشش چرخه عمر عمومی ایجاد نرم افزار

چرخه عمومی نرم افزار از ۳ بخش اصلی تعریف[۶۲]، توسعه[۶۳] و نگهداری[۶۴] تشکیل شده است که هر کدام زیر بخش‌هایی دارند.

تعریف

۱. کاوش قلمرو مسئله و مدلسازی

متدولوژی UP : این متدولوژی از ۴ فاز inception ، elaboration ، construction و transition تشکیل شده است که در هرکدام از این فاز ها iteration هایی انجام می شوند. در فاز inception هدف شناخت مسئله است، از این رو در این فاز جریان های کاری نیازمندی ها و تحلیل پررنگ تر انجام می شوند. با استخراج نیازمندی ها، حوزه سیستم مشخص می شود. بنابراین در فاز inception قلمرو مسئله تا حدی مدلسازی می‌شود. سپس در فاز elaboration با شناخت بیشتر مسئله مدلسازی ساختاری و رفتاری قلمرو مسئله ایجاد و مدلسازی وظیفه ای که در واقع همان use case ها می باشند، کامل تر می شوند و نیازمندی هایی که در این فاز استخراج شدند به آن ها اضافه می شوند. مدلسازی ساختاری برای مدلسازی قلمرو مسئله شامل class diagram ، object diagram و package diagram تحلیل می باشند. مدلسازی رفتاری قلمرو مسئله شامل activity diagram ، sequence diagram و state transition diagram برای کلاس هایی که رفتار وابسته به حالت دارند، می شود.

متدولوژی Booch : این متدولوژی یک متدولوژی تکراری-افزایشی است که در آن فرایند را به عنوان یک فرایند تکرارشونده (micro process) در یک فرایند تکراری در سطح چرخه عمر (macro process) مدلسازی کرده است. macro process به عنوان یک چارچوب کنترلی برای micro process تعریف شده است. macro process از ۵ فاز تشکیل شده است. در هر فاز مراحل micro process بصورت تکراری انجام می شود. در فاز اول بنام مفهوم سازی، نیازمندی های هسته استخراج می شوند و در فاز بعدی یعنی تحلیل، یک مدل از رفتار مطلوب سیستم ایجاد می شود. با استخراج نیازمندی ها قلمرو مسئله مشخص می شود و در تحلیل، قلمرو مسئله مدل می شود. در هر کدام از این فاز ها، مراحل micro process بصورت تکراری اجرا می شوند. در فاز تحلیل، با فعالیت های شناسایی کلاس ها و اشیاء در قلمرو مسئله، شناسایی نحوه تعامل اشیاء در قلمرو مسئله و شناسایی ساختار کلاس ها و اشیاء در قلمرو مسئله، قلمرو مسئله مدل می شود.

---

[۶۲] definition
[۶۳] development
[۶۴] maintenance

متدولوژی OMT : این متدولوژی دارای ۵ فاز است. در فاز اول یعنی فاز تحلیل ابتدا قلمرو مسئله با توجه به نیازمندی‌ها مشخص می شود، سپس ۳ مدل شیء، پویا و وظیفه ای در قلمرو مسئله به صورت تکراری ساخته می شوند. این ۳ مدل در واقع وجوه ساختاری، رفتاری و وظیفه ای سیستم را مدل می کنند. در فازهای بعدی این ۳ مدل تکمیل می شوند و کلاس های قلمرو جواب به آن ها اضافه می شوند.

۲. استخراج نیازمندی ها

متدولوژی UP : همانطور که ذکر شد، متدولوژی UP در فاز inception، نیازمندی های اساسی را ، که حدود ۲۰ درصد کل نیازمندی ها می شوند، استخراج می کند و حوزه سیستم را مشخص می کند. سپس در فاز elaboration تا ۸۰ درصد نیازمندی ها استخراج می شوند و مابقی نیازمندی ها که تا این مرحله قابل استخراج نیستند، در فاز construction استخراج می شوند. این را هم باید ذکر کرد که این متدولوژی use-case-driven است و همه کارها بر مبنای use case ها انجام می شوند. Use case ها نیز در هر مرحله طوری که همه نیازمندی های استخراج شده را پوشش دهند، ساخته می شوند.

متدولوژی Booch : این متدولوژی همانطور که گفته شد، در macro process و در فاز اول آن نیازمندی های هسته را استخراج می کند. باقی نیازمندی ها در طول فرایند کم کم پیدا می شوند و اضافه می شوند.

متدولوژی OMT : این متدولوژی همانطور که گفته شد، دارای ۵ فاز است و در فاز اول یعنی تحلیل، نیازمندی ها را استخراج می کند و قلمرو مسئله را با توجه به نیازمندی ها مدل می کند اما فاز جدا برای استخراج نیازمندی ها در نظر نگرفته است و مهندسی نیازمندی ها انجام نمی دهد.

۳. تحلیل امکانپذیری

متدولوژی UP : این متدولوژی در فاز inception به تحلیل امکانپذیری می پردازد. به این صورت که در جریان کاری تحلیل، بررسی می شود که امکانپذیری وجود دارد یا نه. در همین جهت در جریان کاری طراحی، technical prototyping ، برای بررسی اینکه تکنولوژی مورد نیاز موجود هست یا خیر، و proof of concept prototyping ، برای سنجش نیازمندی ها و بررسی اینکه خواسته کاربر به درستی درک شده است یا خیر، طراحی می شوند. سپس در جریان کاری اجرا، این prototype ها پیاده سازی می‌شوند. همچنین در فاز inception ریسک های مهم نیز شناسایی می شوند. سپس در فاز elaboration امکانپذیری کاملا مشخص می شود و همه ریسک ها شناسایی می شوند. در نتیجه متدولوژی UP به خوبی به تحلیل امکانپذیری می پردازد.

متدولوژی Booch : این متدولوژی به تحلیل امکانپذیری در هیچ یک از فاز هایش اشاره نکرده است.

متدولوژی OMT : این متدولوژی به تحلیل امکانپذیری در هیچ یک از فاز هایش اشاره نکرده است.

توسعه

۱. طراحی معماری

متدولوژی UP : در این متدولوژی در فاز inception یک معماری اولیه سطح بالا تولید می شود تا قابل انجام بودن پروژه مشخص شود، سپس در فاز elaboration معماری کامل و تثبیت می شود که از این پس مبنا کارها می باشد. به این معماری، معماری مبنا[۶۵] قابل اجرا می گویند. همچنین این معماری توسط ذی نفعان نیز باید تایید شود. بنابراین متدولوژی UP طراحی معماری را بصورت کامل انجام می دهد.

متدولوژی Booch : در این متدولوژی، macro process از ۵ فاز تشکیل شده است. در هر فاز مراحل micro process بصورت تکراری انجام می شود. در فاز سوم آن یعنی طراحی، یک معماری برای سیستم ایجاد می‌شود. در واقع در این بخش وارد قلمرو جواب می شویم و هم طراحی معماری و هم بخشی از طراحی تفضیلی سیستم انجام می شوند.

متدولوژی OMT : این متدولوژی دارای ۵ فاز است که در فاز دوم یعنی طراحی سیستم، ساختار سطح بالای سیستم را مشخص می کند و ویژگی های قلمرو جواب را اضافه می کند. در واقع در این فاز طراحی معماری صورت می‌گیرد و طراحی معماری را به عنوان یک مرحله جدا از طراحی تفضیلی تعریف کرده است. (توضیحات بیشتر در بخش ۲-۳)

۲. طراحی تفضیلی: در این بخش اجزاء سیستم و رفتار هایشان با جزئیات مشخص می شود و چگونگی تعامل شیء ها با هم برای تحقق نیازمندی ها نشان داده می شود.

متدولوژی UP : این متدولوژی در فاز elaboration در جریان کاری طراحی، کمی وارد طراحی تفضیلی می‌شود، در حدی که جزئیات پیاده سازی use case های ریسکی اضافه می شود و در جریان کاری اجرا و تست، این use case ها پیاده سازی و تست می شوند. مابقی و در واقع بخش اصلی طراحی تفضیلی در فاز construction و در جریان کاری طراحی، انجام و تکمیل می شود؛ بطوری که اجزاء سیستم و رفتارهایشان با جزئیات مشخص می شود و جزئیات پیاده سازی به آن ها اضافه می شود.

[۶۵] Executable architecture baseline

متدولوژی Booch : در این متدولوژی همانطور که ذکر شد، در فاز سوم macro process ، طراحی تفضیلی شروع می‌شود و کلاس ها و اشیائی که در قلمرو جواب هستند اضافه می شوند و نحوه تعامل آن ها مشخص می‌شود. مابقی طراحی تفضیلی در فاز ۴ام یعنی تکامل، انجام می شود.

متدولوژی OMT : این متدولوژی از ۵ فاز تشکیل شده است که در فاز سوم یعنی طراحی شیء، طراحی تفضیلی انجام می دهد. به این صورت که در این فاز تمام جزئیات کلاس ها، روابط بین آن ها، خواص[۶۶] آن‌ها، عملکرد[۶۷] هایشان با توجه به معماریی که طراحی شده است، مشخص می شود. همچنین ساختار های داده و شیء های داخلی نیز مشخص می شود. در واقع بیشتر جزئیاتی که برای پیاده سازی نیاز هست باید در این فاز مشخص شود.

۳. برنامه نویسی

متدولوژی UP : در این متدولوژی بخش عمده پیاده سازی در فاز construction انجام می شود. در این فاز بصورت تکراری-افزایشی use case ها پیاده سازی می شوند و معماری مبنا قابل اجرا که در فاز elaboration ساخته شده بود، تکامل پیدا می کند و به سیستم نهایی تبدیل می شود. البته در فازهای دیگر نیز کمی پیاده سازی صورت می گیرد. در فاز inception ، prototype هایی برای تحلیل امکانپذیری پیاده سازی می شوند. در فاز elaboration ، معماری مبنا قابل اجرا پیاده سازی می شود. همچنین در این فاز use case های ریسکی نیز باید پیاده سازی بشوند یا کد proof-of-concept برای نشان دادن امکانپذیری پیاده سازی آن ها، تولید شود. و در فاز transition ، سیستم که در محیط کاربر اجرا می‌شود ممکن است ایراداتی داشته باشد، در همین جهت برای برطرف کردن ایرادات کمی پیاده سازی انجام می‌شود.

متدولوژی Booch : در این متدولوژی، macro process از ۵ فاز تشکیل شده است. در هر فاز مراحل micro process بصورت تکراری انجام می شود. در فاز ۴ ام پیاده سازی، تست و استقرار سیستم انجام می شوند.

متدولوژی OMT : این متدولوژی ۵ فاز دارد که فاز چهارم، فاز کد زدن یا همان برنامه نویسی می باشد که سورس کد را تولید می کند.

۴. تست

متدولوژی UP : در این متدولوژی تست در فازهای construction و transition پررنگ تر می باشد، اما در فاز elaboration نیز کمی تست انجام می شود. در فاز elaboration ، معماری مبنا قابل اجرا بعد از طراحی

[۶۶] attributes
[۶۷] operation

و پیاده سازی، تست می شود. در فاز construction نسخه ای که تا این فاز کامل . پیاده سازی شده است، در محیط اجرا تست می شود. در فاز transition نسخه ای از نرم افزار که در فاز قبلی ساخته شده بود، به محیط کاربر منتقل می شود و در آن جا روی آن تست آلفا، تست بتا، تست سیستم و تست پذیرش نهایی انجام می شود و در صورت بروز مشکل، برای برطرف کردن آن ها کمی وارد جریان های کاری پیاده سازی، طراحی و در صورت نیاز تحلیل می شویم. در تست آلفا، یک محیط شبیه سازی شده از محیط کاربر در محیط اجرا ساخته می شود و کاربر آن را اجرا می کند و فیدبک می دهد. در تست بتا نرم افزار برای مشتریان بالقوه فرستاده می شود و کاربران در محیط خود نصب می کنند و فیدبک می‌دهند. در تست سیستم، تست های سیستمی یکبار در محیط اجرا و یکبار در محیط کاربر انجام می‌شوند. و تست پذیرش در نهایت که سیستم بطور کامل در محیط کاربر نصب شد، انجام می شود.

متدولوژی Booch : در این متدولوژی همانطور که ذکر شد، در فاز ۴ ام macro process ، فعالیت ها در راستای تست سیستم انجام می شوند.

متدولوژی OMT : این متدولوژی ۵ فاز دارد که فاز آخر آن، تست می باشد که سناریو های تست در آن ایجاد می شوند. بنابراین این بخش از چرخه عمر نرم افزار را نیز پوشش می دهد.

۵. استقرار[۶۸]: استقرار به معنی قرار دادن محصول در محیط کاربر می باشد.

متدولوژی UP : در این متدولوژی با اینکه استقرار به عنوان یک جریان کاری جدا، مثل RUP ، در نظر گرفته نشده است، اما در طی جریان های کاری اجرا و تست انجام می شود. در انتهای فاز construction تمهیدات لازم برای استقرار صورت می گیرد و در فاز transition ، محیط کاربر برای نصب نرم افزار آماده می‌شود، روی نرم افزار تغییراتی در جهت آماده سازی آن برای انتقال به محیط کاربر صورت می گیرد. و درنهایت نرم افزار بطور کامل به محیط کاربر منتقل می شود.

متدولوژی Booch : این متدولوژی همانطور که ذکر شد، در فاز ۴ ام macro process ، علاوه بر پیاده سازی و تست، سیستم در محیط کاربر قرار می گیرد.

متدولوژی OMT : این متدولوژی فاقد مرحله ی استقرار یا قرار دادن نرم افزار در محیط کاربر می باشد.

[۶۸] deployment

نگهداری: از انواع آن می توان به نگهداری تصحیحی[۶۹]، تکمیلی[۷۰]، تطبیقی[۷۱] و اجتنابی[۷۲] اشاره کرد.

متدولوژی UP : در این متدولوژی فازی جدا برای نگهداری در نظر گرفته نشده است اما در انتهای فاز transition، باید تیم نگهداری شکل بگیرد، آموزش به آن ها صورت گیرد و فرایند نگهداری مشخص شود. همچنین تمهیدات لازم برای ساپورت سیستم و مشاوره دادن به کاربر نیز انجام شود و تیم ساپورت برای اینکار آموزش ببینند. سپس اگر ساخت سیستم تمام شده بود وارد مرحله نگهداری می شویم و در صورتی که قرار بر ساخت نسخه های بعدی سیستم بود، فعالیت های توسعه نسخه جدید و نگهداری نسخه های قبلی بطور موازی انجام می‌شوند.

باید اشاره کرد که این متدولوژی خود فرایندی برای نگهداری تعریف نکرده است و برای نگهداری همان تکرار فازها و جریان های کاری موجود در هر فاز را پیشنهاد داده است. اما می دانیم که ماهیت مشکلاتی که در نگهداری پیش می آیند غیر قابل پیش‌بینی است، بنابراین نمی توانیم از فرایند تکراری این متدولوژی استفاده کنیم و باید فرایند دیگری برای نگهداری تعریف شود. در نتیجه متدولوژی UP مرحله نگهداری را پوشش نمی دهد.

متدولوژی Booch : در این متدولوژی، macro process از ۵ فاز تشکیل شده است. در هر فاز مراحل micro process بصورت تکراری انجام می شود. در فاز ۵ ام فعالیت های مربوط به نگهداری سیستم انجام می شوند.

متدولوژی OMT : در این متدولوژی هیچ اشاره ای به مرحله نگهداری نرم افزار نشده است.

مقایسه متدولوژی UP و Booch :

شباهت ها:

هر دو متدولوژی فعالیت های کاوش و مدلسازی قلمرو مسئله و استخراج نیازمندی ها از بخش تعریف، و بخش توسعه به طور کامل یعنی طراحی معماری، طراحی تفضیلی، پیاده سازی، تست و استقرار را پوشش می دهند. بنابراین هیچ یک چرخه عمر عمومی نرم افزار را بطور کامل پوشش نمی دهند.

تفاوت ها:

متدولوژی UP فعالیت تحلیل امکانپذیری از تعریف را پوشش می دهد در حالی که متدولوژی Booch آن را ندارد. متدولوژی Booch فاز نگهداری را پوشش می دهد، در حالی که متدولوژی UP آن را ندارد.

---

[۶۹] corrective
[۷۰] perfective
[۷۱] adaptive
[۷۲] prevented

مقایسه:

همانطور که در بخش تفاوت ها ذکر شد متدولوژی UP از لحاظ داشتن فعالیت تحلیل امکانپذیری نسبت به متدولوژی Booch برتری دارد و متدولوژی Booch به دلیل داشتن فاز نگهداری نسبت به متدولوژی UP برتری دارد.

مقایسه متدولوژی UP و OMT :

شباهت ها:

هر دو متدولوژی فعالیت های کاوش و مدلسازی قلمرو مسئله و استخراج نیازمندی ها از بخش تعریف، و طراحی معماری، طراحی تفضیلی، پیاده سازی و تست از بخش توسعه را پوشش می دهند. هیچ یک از دو متدولوژی فازی برای نگهداری در نظر نگرفته اند. بنابراین هیچ یک چرخه عمر عمومی نرم افزار را بطور کامل پوشش نمی دهند.

تفاوت ها:

متدولوژی UP فعالیت های تحلیل امکانپذیری از بخش تعریف و استقرار از بخش توسعه را پوشش می دهد در حالی که متدولوژی OMT آن ها را ندارد. همچنین متدولوژی OMT با اینکه نیازمندی ها را استخراج می کند اما در فازی جداگانه به آن نمی پردازد.

مقایسه:

متدولوژی UP چرخه عمر عمومی نرم افزار را بهتر از متدولوژی OMT پوشش می دهد. متدولوژی OMT فعالیت های تحلیل امکانپذیری و استقرار را پوشش نمی دهد در حالی که متدولوژی UP آن ها را دارد.

۴-۱-۳- پشتیبانی از فعالیت های چتری

فعالیت های چتری به فعالیت هایی گفته می شود که در کل طول چرخه عمر نرم افزار برای مدیریت ریسک، مدیریت پروژه و تضمین کیفیت صورت می گیرند.

- مدیریت ریسک: به این معناست که ارزیابی ریسک و کاهش ریسک را در چرخه ی عمر پوشش دهیم. یعنی امکان پذیری[۷۳] کار ها را از ابتدا بررسی کنیم و کارهایی که ریسک پذیر هستند را در ابتدا انجام دهیم (risk-based planning). تکنیک های تحلیل امکان سنجی اولیه[۷۴]، نمونه سازی اولیه[۷۵]، برنامه‌ریزی

[۷۳] feasibility
[۷۴] preliminary feasibility analysis
[۷۵] prototyping

مبتنی بر ریسک[۷۶]، فرآیند تکراری-افزایشی[۷۷]، مشارکت فعال کاربر[۷۸]، انتشار زودهنگام[۷۹]، V&V (validation[۸۰] & verification[۸۱]) مداوم، بررسی های تکراری[۸۲]، و یکپارچه سازی مداوم[۸۳] برای مدیریت ریسک مفید هستند.

- مدیریت پروژه: تکنیک های برنامه ریزی، زمان بندی و کنترل باید انجام شوند؛ برنامه ها باید به طور مکرر مورد بازبینی و تجدید نظر قرار گیرند و باید به مدیریت تیم و تقویت همکاری اعضای درون تیم و بیرون تیم توجه شود.
- تضمین کیفیت: فعالیت ها شامل ارزیابی کیفیت و تکنیک های بهبود باشند. تکنیک‌های مرور فنی تکراری[۸۴]، طراحی بر اساس قرارداد، V&V مستمر و تکنیک‌هایی که قابلیت ردیابی نیازمندی‌ها[۸۵] را افزایش می‌دهند، مفید می باشند.

متدولوژی UP : متدولوژی UP دارای ۴ فاز inception ، elaboration ، construction و transition می‌باشد که در هرکدام از این فاز ها iteration هایی انجام می شوند. هر iteration شامل ۵ جریان کاری نیازمندی ها، تحلیل، طراحی، اجرا و تست است.

- مدیریت ریسک:

✓ در فاز inception ریسک های بحرانی شناسایی می شوند و تحلیل امکانپذیری انجام می شود. به این صورت که technical prototyping و proof of concept prototyping برای بررسی وجود تکنولوژی مورد نیاز و سنجش نیازمندی ها، انجام می شوند. این prototype ها دورریختنی هستند و برای سنجش امکانپذیری ساخته می شوند.

✓ در فاز inception علاوه بر شناسایی ریسک های بحرانی، سند ارزیابی ریسک نیز توسط مدیر پروژه تولید می شود. در فاز elaboration سند ارزیابی ریسک اصلاح می شود و برای use case های حاوی ریسک طراحی تفضیلی انجام می شود. همچین این use case ها پیاده سازی و تست نیز می شوند. البته ممکن است به جای پیاده سازی کامل، کد proof-of-concept برای بررسی

---

[۷۶] risk-based planning
[۷۷] iterative-incremental process
[۷۸] active user involvement
[۷۹] early release
[۸۰] اعتبار سنجی: سنجیدن اینکه سیستم مورد تایید کاربر هست یا خیر
[۸۱] صحت سنجی: بررسی اینکه سیستم درست کار می کند یا خیر
[۸۲] iterative reviews
[۸۳] continuous integration
[۸۴] iterative technical reviews
[۸۵] requirements traceability

قابل پیاده سازی بودن آن ها تولید شود. در واقع در انتهای این فاز ریسک ها باید بطور کامل شناسایی شوند.

- ✓ در طول اجراء فازها هرگاه که project plan اصلاح می شود، بررسی می شود که امکانپذیری‌هایی که مشخص شده است را نقض نکنند.
- ✓ فرایند متدولوژی UP تکراری-افزایشی است که به مدیریت ریسک کمک می کند.
- ✓ جریان های کاری که بصورت تکراری در هر فاز انجام می شوند شامل تست نیز می باشد. هر use case در هر فازی که تولید می شود، تست نیز می شود. همچنین در هر فاز محصولاتی که تولید می شوند، اعم از project plan ، طرح توجیه اقتصادی، معماری و ... به ذی نفعان نشان داده می شود و تایید آن ها گرفته می شود. بنابراین هم کاربر مشارکت فعال دارد و هم V&V (اعتبار سنجی و صحت سنجی) بصورت مستمر انجام می شود.
- ✓ در جریان کاری تست که در هر فاز در تکرار ها انجام می شود، یکپارچه سازی نیز صورت می‌گیرد.
- ✓ در هر بار اجرای ۴ فاز ذکر شده، یک نسخه تولید می شود و برای تولید نسخه های بعدی فرایند دوباره از فاز inception اجرا می شود.

➢ مدیریت پروژه:

- ✓ در متدولوژی UP جریان کاری مدیریت پروژه که در متدولوژی RUP وجود داشت، حذف شده است، اما این متدولوژی فعالیت های مدیریت پروژه را در طی فازها انجام می دهد. در فاز inception فعالیت‌های برنامه ریزی و مدیریت پروژه شامل نمودار های گانت[۸۶]، برنامه ها[۸۷]، بودجه ها و غیره، انجام می شوند. البته برنامه ها می توانند iteration های ابتدایی فاز elaboration را شامل شوند و برنامه ریزی مابقی iteration های فاز elaboration، در طی این فاز کامل می شود. در انتهای فاز elaboration برنامه دقیق برای iteration های اولیه فاز بعدی یعنی فاز construction، ایجاد می‌شود و برنامه ریزی مابقی iterationها در طول این فاز کامل می شود. بنابراین این برنامه ریزی ها بطور مکرر در طی فازها مورد بازبینی و تجدید نظر قرار می گیرند. البته این متدولوژی به مدیریت افراد اشاره ای نکرده است اما مدیریت پروژه را به خوبی انجام می دهد.

[۸۶] Gantt charts
[۸۷] plans

- تضمین کیفیت:
  - ✓ همانطور که ذکر شد در متدولوژی UP در هر فاز جریان های کاری تکرار می شوند که حاوی جریان کاری تست می باشد. که علاوه بر اینکه موجب انجام V&V مستمر می شود، بطور تکراری محصولات بررسی می شوند و کیفیت آن ها تضمین می شود.
  - ✓ این متدولوژی مبتنی بر use case است و همه فعالیت ها بر اساس use case ها انجام می شوند. بنابراین ردیابی به نیازمندی ها بطور مستقیم وجود دارد.
  - ✓ در فاز elaboration معیار های کیفی مثل نرخ کشف نقص[۸۸] و تراکم نقص قابل قبول[۸۹]، تعریف می شوند که کیفیت کد را نشان می دهند و با پایین نگه داشتن این معیار ها می توان کیفیت را تضمین کرد.

متدولوژی Booch :

- متدولوژی Booch اشاره ای به کارهایی که در جهت مدیریت ریسک انجام می شوند، نکرده است. برای مثال هیچ یک از تکنیک های تحلیل امکان سنجی اولیه، نمونه سازی اولیه، برنامه ریزی مبتنی بر ریسک، مشارکت فعال کاربر و بررسی های تکراری انجام نمی‌شوند. اما فرایند این متدولوژی، یک فرایند تکراری-افزایشی است. در هر فاز macro process ، که یک چارچوب مدیریتی برای macro process است، micro process بصورت تکراری انجام می شود و همچنین V&V مستمر صورت می گیرد و این ها مدیریت ریسک را در صورت بروز آن ساده تر می‌کند.
- در این متدولوژی فعالیت های مدیریتی در macro process انجام می شوند. برای مثال فعالیت هایی مثل تضمین کیفیت، مرور کد و مستندات در این بخش انجام می شوند. همچنین در macro process ، برنامه ریزی، زمانبندی و کنترل نیز انجام می شود. اما Booch اشاره ای به مدیریت اعضای تیم نکرده است. اما در کل مدیریت پروژه در این متدولوژی در سطح خوبی قرار دارد.
- در متدولوژی Booch همانطور که ذکر شد، فعالیت هایی مثل تضمین کیفیت، مرور کد و مستندات در macro process انجام می شوند. بنابراین مرور کد بصورت تکراری انجام می شود و همچنین کیفیت مد نظر کاربر چک می شود. بنابراین این متدولوژی تضمین کیفیت دارد.

متدولوژی OMT :

- متدولوژی OMT اشاره ای به کارهایی که در جهت مدیریت ریسک انجام می شوند، نکرده است. برای مثال هیچ یک از تکنیک های تحلیل امکان سنجی اولیه، نمونه سازی اولیه، برنامه ریزی مبتنی بر ریسک،

[۸۸] Defect discovery rate
[۸۹] Acceptable defect densities

فرآیند تکراری-افزایشی، مشارکت فعال کاربر، انتشار زودهنگام، بررسی های تکراری، و یکپارچه سازی مداوم انجام نمی‌شوند. اما سه مدل شیء، پویا و وظیفه ای پس از ساخته شدن و اصلاح شدن، صحت‌سنجی می شوند که می‌تواند کمی به مدیریت ریسک کمک کند.

- در این متدولوژی برنامه ریزی، زمانبندی و کنترل انجام نمی شوند. اشاره ای هم به استفاده از تکنیک‌هایی مثل بازبینی مکرر برنامه ها و مدیریت تیم نشده است.
- فعالیت ها در OMT حاوی ارزیابی کیفیت نمی باشند. از تکنیک هایی مثل مرور فنی تکراری، طراحی بر اساس قرارداد و تکنیک‌هایی که قابلیت ردیابی نیازمندی‌ها را افزایش می‌دهند، استفاده ای نشده است. تنها کاری که صورت می گیرد صحت سنجی مدل های شیء، پویا و وظیفه ای است که می تواند کمی به تضمین کیفیت کمک کند. البته این متدولوژی حاوی فاز تست نیز می باشد که بعد از پیاده سازی سیستم انجام می شود که آن نیز کمک کننده است، اما تأثیری در تضمین کیفیت تا قبل از پیاده سازی سیستم ندارد.

مقایسه متدولوژی UP و Booch :

شباهت ها:

- ✓ هر دو متدولوژی فرایند تکراری-افزایشی دارند که به مدیریت ریسک کمک می کند. هر دو نیز در فعالیت‌های تکراریشان تست انجام می دهند که در آن V&V مستمر انجام می شود.
- ✓ هر دو متدولوژی فعالیت های برنامه ریزی، زمانبندی و کنترل را انجام می دهند بنابراین مدیریت پروژه صورت می گیرد اما در هیچ یک به مدیریت افراد درگیر پروژه اشاره نشده است.
- ✓ هر دو متدولوژی فعالیت هایی برای تضمین کیفیت انجام می دهند و سطح آن در هر دو بالاست.

تفاوت ها:

- ✓ متدولوژی UP از تکنیک های تحلیل امکان سنجی اولیه، نمونه سازی اولیه، برنامه‌ریزی مبتنی بر ریسک و مشارکت فعال کاربر که برای مدیریت ریسک مفید هستند، استفاده می کند. در حالی که متدولوژی Booch از هیچ کدام از آن ها استفاده نمی کند.
- ✓ در متدولوژی UP، plan به طور مستمر مورد مرور و بازنگری قرار می گیرد در حالی که در متدولوژی Booch اشاره ای به آن نشده است.
- ✓ متدولوژی UP ردیابی به نیازمندی ها را بطور مستقیم دارد و همچنین معیار هایی نیز برای سنجش کیفیت تعریف کرده است. در حالی که متدولوژی Booch این ها را ندارد.

مقایسه:

متدولوژی UP کاملا مدیریت ریسک را انجام می دهد و از این جهت نسبت به متدولوژی Booch برتری دارد. هر دو متدولوژی مدیریت پروژه و تضمین کیفیت را انجام می دهند اما متدولوژی UP به دلیل تفاوت‌هایی که ذکر شد بهتر عمل می کند و نسبت به متدولوژی Booch برتری دارد.

مقایسه متدولوژی UP و OMT :

شباهت ها: –

تفاوت ها:

- ✓ متدولوژی UP از تکنیک های تحلیل امکان سنجی اولیه، نمونه سازی اولیه، برنامه‌ریزی مبتنی بر ریسک، مشارکت فعال کاربر، V&V مستمر و یکپارچه سازی مداوم که برای مدیریت ریسک مفید هستند، استفاده می‌کند. در حالی که متدولوژی OMT از هیچ یک از آن ها استفاده نمی کند.
- ✓ متدولوژی UP فعالیت های برنامه ریزی، زمانبندی و کنترل را انجام می دهند بنابراین مدیریت پروژه صورت می‌گیرد. همچنین در آن plan به طور مستمر مورد مرور و بازنگری قرار می گیرد.
- ✓ متدولوژی UP، V&V مستمر و ردیابی به نیازمندی ها را بطور مستقیم دارد و همچنین معیار هایی نیز برای سنجش کیفیت تعریف کرده است. در حالی که در متدولوژی OMT تنها کاری که انجام می شود صحت سنجی مدل های شیء، پویا و وظیفه ای و همچنین فاز تست است که می تواند کمی به تضمین کیفیت کمک کند.

مقایسه:

متدولوژی UP به خوبی از فعالیت های چتری یعنی مدیریت ریسک، مدیریت پروژه و تضمین کیفیت پشتیبانی می کند در حالی که متدولوژی OMT هیچ یک از آن ها را ندارد و این یکی از برتری های متدولوژی UP نسبت به آن است.

۴-۱-۴- بی درزی[۹۰] و همواری انتقال[۹۱] از یک محصول به محصول دیگر

بی درزی به این معناست که در زنجیره ی مدلسازی و کارها تغییرات پارادایم[۹۲] نداشته باشیم. ۲ تکنیک برای برای اینکه درز نداشته باشیم این است که همه وظایف و محصولات را بر یک مفهوم مشترک پایه گذاری کنیم یا همه ی کار ها را بر مبنای ثابتی از مدل ها پیش ببریم.

همچنین باید بین فاز ها،مرحله ها[۹۳] و وظیفه ها[۹۴] انتقال هموار باشد. ساخت محصولات جدید از روی محصولات موجود تهدیدی برای همواری انتقال است.

متدولوژی UP : در این متدولوژی نیازمندی ها در use case ها پوشش داده می شوند و همه فعالیت ها و محصولات بر مبناء آن ها پیش می روند. بنابراین این متدولوژی use-case-driven است و همه وظایف و محصولات بر یک مفهوم مشترک پایه گذاری شده‌اند در نتیجه تا حدی بی درز است، اما use case یک نمودار غیر شیء گراست درحالی که سایر محصولات شیء گرا هستند. بنابراین ساخت محصولات، مانند sequence diagram ، از روی use case ها پارادایم ایجاد می کند و این به بی درز بودن UP کمی خدشه وارد می کند و نمی توان گفت کاملا بی درز است.

در این متدولوژی جریان های کاری بصورت تکراری در فازها انجام می شوند و فرایند این متدولوژی تکراری-افزایشی است. بنابراین فعالیت ها کم کم انجام می شوند و ناهمواری بین آن ها ایجاد نمی شود. همچنین محصولات با زبان مدلسازی UML ساخته می شوند که در طی فاز ها کامل می شوند و ناهمواری ایجاد نمی کنند.

متدولوژی Booch : این متدولوژی یک متدولوژی تکراری-افزایشی است که در آن macro process به عنوان یک چارچوب کنترلی برای micro process تعریف شده است. macro process از ۵ فاز تشکیل شده است. در هر فاز micro process بصورت تکراری انجام می شود. ۵ فاز آن عبارتند از مفهوم‌سازی، تحلیل، طراحی، تکامل و نگهداری. خود micro process از ۴ مرحله تشکیل شده است که بصورت تکراری انجام می‌شوند. در این مراحل نمودارهایی بسته به سطح انتزاعی که هستیم تولید یا تکمیل می شوند و جزئیات پیاده سازی به آن ها اضافه می‌شود. بنابراین به این دلیل که مدلسازی بصورت تدریجی و تکراری انجام می شود، بی درزی و انتقال هموار بین فازها وجود دارد. البته باید ذکر کرد که ساخت مدل های شیء گرا از روی مستندات نیازمندی غیر شیء گرا یک پارادایم ایجاد می‌کند که به بی درزی آسیب می زدند.

---

[۹۰] seamlessness
[۹۱] smoothness of transition
[۹۲] paradigm shift
[۹۳] stages
[۹۴] tasks

متدولوژی OMT : این متدولوژی در همان فاز اولش یعنی تحلیل، ۳ مدل شیء،پویا و وظیفه ای می سازد و در بقیه ی فاز ها بر مبنای همین ۳ مدل عمل می کند و همان ها را تکمیل می کند. بنابراین همه ی محصولات و وظایف بر مبنای یک مفهوم بنا نهاده می شوند. اما مشکلی که وجود دارد این است که OMT در مدل وظیفه ای از نمودار DFD استفاده می کند که این نمودار بر مبنای شیء گرایی نیست. درنتیجه درز ایجاد می شود.

اما با توجه به ساخت ۳ مدل در ابتدای کار و تکمیل آن ها در مراحل بعدی، OMT انتقال هموار دارد.

مقایسه متدولوژی UP و Booch :

شباهت ها:

هر دو متدولوژی فرایند تکراری-افزایشی دارند و محصولات به مرور ایجاد و تکمیل می شوند، بنابراین هر دو متدولوژی انتقال هموار دارند و تا حدی بی درز هستند. اما در هر دو از روی نیازمندی ها که بصورت غیر شیءگرا ثبت شدند، مدل های شیء گرا ایجاد می شوند، بنابراین یک پارادایم وجود دارد که به بی درزی آسیب می‌زند.

تفاوت ها:

متدولوژی UP ، use-case-driven است یعنی همه مدل ها بر مبنای use case ها و در واقع نیازمندی ها ساخته می شوند. در حالی که متدولوژی Booch، requirements-driven نیست.

مقایسه:

متدولوژی UP به دلیل use-case-driven بودن و درنتیجه اینکه همه چیز برمبنای یک مفهوم ثابت است، از لحاظ بی درزی بهتر از متدولوژی Booch عمل می کند.

مقایسه متدولوژی UP و OMT :

شباهت ها:

هر دو متدولوژی مدل ها را به تدریج ایجاد و تکمیل می کنند بنابراین انتقال هموار دارند. متدولوژی UP مدل‌ها را بر پایه یک مفهوم مشترک (use case) می سازد و متدولوژی OMT ۳ مدلی که در ابتدا ساخته را تکمیل می کند. بنابراین هر دو تا حدی بی درز هستند، اما در هر دو متدولوژی مدل های شیء گرا از روی نیازمندی ها که بصورت غیر شیء گرا مستند یا مدل شدند، ساخته می شوند که این یک پارادایم ایجاد می کند و به بی درزی آسیب می زند.

تفاوت ها:

در متدولوژی UP همه مدل ها جز use case ها، شیء گرا هستند اما در متدولوژی OMT ، نمودار DFD نیز وجود دارد که یک پارادایم دیگر ایجاد می کند.

مقایسه:

همانطور که گفته شد هر دو متدولوژی به بی درز بودنشان خدشه وارد شده است اما این آسیب در متدولوژی OMT به دلیل وجود نمودار DFD بیشتر است. بنابراین از این نظر متدولوژی UP برتری دارد.

۴-۱-۵- مبتنی بر نیازمندی ها (وظیفه ای و غیر وظیفه ای)

نیازمندی های وظیفه ای و غیر وظیفه ای باید زود در فرایند مشخص شوند، به طور مستقل مدلسازی شوند و مبنا برای طراحی و اجرا قرار گیرند.

متدولوژی UP : در این متدولوژی در فاز inception نیازمندی های اساسی، که حدود ۲۰ درصد کل نیازمندی‌ها هستند، استخراج می شوند و در use case ها مدل می شوند سپس در فاز elaboration تا ۸۰ درصد نیازمندی‌ها استخراج می شوند و مابقی نیازمندی ها که تا این مرحله قابل استخراج نیستند، در فاز construction استخراج می شوند. این متدولوژی use-case-driven است و همه کارها بر مبنای use case ها انجام می شوند. بنابراین این متدولوژی کاملا مبتنی بر نیازمندی های وظیفه ای است. همچنین در فاز inception یک سند چشم انداز تهیه می شود که شامل نیازمندی های وظیفه ای و غیر وظیفه ای و محدودیت هایی است که از سمت کاربر تعیین می‌شوند. بنابراین متدولوژی UP بر مبنای نیازمندی های غیر وظیفه ای نیز می باشد.

متدولوژی Booch : در این متدولوژی، macro process از ۵ فاز تشکیل شده است. در هر فاز مراحل micro process بصورت تکراری انجام می شود. در فاز اول یعنی مفهوم سازی، نیازمندی های هسته استخراج می شوند. بنابراین یک فاز جداگانه برای استخراج نیازمندی ها در نظر گرفته شده است. البته در این بخش تنها ۲۰ درصد نیازمندی ها می‌توانند پیدا شوند و مابقی در طی اجرای فرایند نمود پیدا می کنند. قلمرو مسئله بر اساس نیازمندی های اولیه مشخص می شود اما از آن پس از سناریو هایی برای بررسی مبتنی بر نیازمندی بودن استفاده نمی شود. همچنین نیازمندی ها بطور مستقل مدلسازی نمی شوند. بنابراین این متدولوژی requrements-driven نیست اما مبتنی بر نیازمندی های وظیفه ای می باشد. در این متدولوژی به همه نیازمندی های غیر وظیفه ای توجه نمی شود، اما نیازمندی های غیر وظیفه ای مثل کیفیت، کامل بودن و زمانبندی در macro process مورد تمرکز قرار می گیرد. همچنین این متدولوژی نیز در فازهای خود استقرار و نگهداری دارد که نوعی نیازمندی غیر وظیفه ای هستند.

متدولوژی OMT: این متدولوژی ۵ فاز دارد که در فاز اول یعنی تحلیل، نیازمندی های کاربران، توسعه دهندگان و مدیران را استخراج می کند. اما با وجود این به تحلیل نیازمندی ها بطور واضح اشاره نشده است و نیازمندی ها بطور مستقل مدلسازی نمی شوند. بنابراین نمی توان گفت که کاملا بر مبنای نیازمندی ها می باشد. اما OMT در فاز طراحی سیستم اولویت های trade-off را مشخص می کند که این یعنی نیازمندی های غیر وظیفه ای را در نظر می‌گیرد.

مقایسه متدولوژی UP و Booch :

شباهت ها:

هر دو متدولوژی بر مبنای نیازمندی های وظیفه ای ساخته می شوند.

تفاوت ها:

متدولوژی UP ، requirements-driven است در حالی که متدولوژی Booch نیست. همچنین در متدولوژی UP نیازمندی های غیر وظیفه ای در قالب سند چشم انداز مورد توجه قرار می گیرد در حالی که متدولوژی Booch خیلی اشاره ای به نیازمندی های غیر وظیفه ای نکرده است.

مقایسه:

همانطور که در بخش تفاوت ها ذکر شد، متدولوژی UP توجه بیشتری به نیازمندی های غیر وظیفه ای دارد و همچنین requirements-driven است و از این نظر ها نسبت به متدولوژی Booch برتری دارد.

مقایسه متدولوژی UP و OMT :

شباهت ها:

هر دو متدولوژی بر مبنای نیازمندی های وظیفه ای و غیر وظیفه ای می باشند.

تفاوت ها:

متدولوژی OMT فازی جداگانه برای استخراج نیازمندی ها در نظر نگرفته است و نیازمندی ها را بطور جداگانه مدل نمی کند. بنابراین نمی توان گفت که به طور کامل مبتنی بر نیازمندی است. در حالی که متدولوژی UP use-case-driven است و همه چیز بر مبنای نیازمندی ها ساخته می شود.

مقایسه:

با توجه به توضیحات ذکر شده، متدولوژی UP کاملا بر مبنای نیازمندی هاست و از این نظر نسبت به متدولوژی OMT برتری دارد.

۴-۱-۶- آزمون پذیری، ملموس بودن، قابل ردیابی به نیازمندی ها

- آزمون پذیری

آزمون پذیری محصولات به پیچیدگی آن ها وابسته است بنابراین محصولات باید ساده، کم و قابل فهم باشند و روابط بین آن ها حداقل باشد. همچنین محصولات باید یکدیگر را در زمینه فرایند کامل کنند نه اینکه باعث پیچیده شدن یکدیگر بشوند.

متدولوژی UP : در این متدولوژی با اینکه محصولات خیلی پیچیدگی ندارند اما تعداد آن ها کم نیست و وابستگی بین آن ها زیاد است. برای مثال sequence diagram و communication diagram هم ریخت هستند و این باعث می شود که هر جا تغییری ایجاد می کنیم در آن یکی هم تغییر را منتشر کنیم. در UP هیچ تمهیدی برای تعدد محصولات و وابستگی بین آن ها ایجاد نشده است. بنابراین متدولوژی UP آزمون پذیر نیست.

متدولوژی Booch : در این متدولوژی، در micro process که فعالیت های ریزدانه ایجاد نرم افزار انجام می‌شوند محصولاتی تولید می شوند. در مرحله اول که کلاس ها و اشیاء شناسایی می شوند، دیکشنری داده تولید می شود. در مرحله دوم که معناشناسی کلاس ها و اشیاء مشخص می شود، نمودار های حالت، نمودارهای شیء و نمودارهای تعامل ساخته می شوند. در مرحله سوم که روابط کلاس ها و اشیاء شناسایی می‌شوند، نمودارهای کلاس، نمودار های ماژول و نمودار های شیء، ایجاد می شوند. و در مرحله آخر که رابطها شناسایی می شوند و کلاس ها و اشیاء پیاده سازی می شوند، نمودارهای فرایند تولید می شوند. این محصولات ساده هستند و بطور تکراری در طی مراحل ساخته و تکمیل می شوند. بنابراین می توان گفت که این متدولوژی آزمون پذیر است.

متدولوژی OMT : این متدولوژی دارای ۵ فاز است که در فاز اول یعنی تحلیل ۳ مدل شیء، پویا و وظیفه ای ساخته می شوند که در هر کدام محصولاتی تولید می شوند(class diagram و data dictionary در مدل شیء، Event-Trace Diagrams و State Transition Diagrams در مدل پویا و نمودارهای DFD در مدل وظیفه ای). در مابقی فاز ها به این مدل ها جزئیات اضافه می شود و نمودار ها تکمیل می شوند. در نتیجه محصولات ساده و قابل فهم هستند و همچنین تعداد آن ها نیز کم می باشد. این ۳ مدل کاملا از هم مستقل

نیستند و وابستگی هایی دارند اما این وابستگی ها در راستای تکمیل مدل ها صورت می گیرد. در نتیجه می‌توان گفت این متدولوژی آزمون پذیر است.

- ملموس بودن

ملموس بودن محصولات به مخاطب بستگی دارد. از دید کاربر محصولات باید قابل اجرا باشند و نحو و معنای آن ها برای کاربر قابل فهم باشد. از دید توسعه دهنده محصولاتی که ایجاد می شوند باید در فرایند، مفید باشند.

متدولوژی UP : در این متدولوژی محصولاتی مثل سند چشم انداز، طرح توجیه اقتصادی و ... تولید می‌شوند که فاقد پیچیدگی هستند و برای کاربر ملموس اند. نمودار ها با زبان مدلسازی UML مدل می‌شوند که اکثرا پیچیدگی ندارند اما برای مثال با اینکه class diagram تحلیل برای کاربر ملموس است اما class diagram طراحی ملموس نیست. همچنین کاربرد اکثر محصولات برای توسعه دهندگان مشخص است اما وجود نمودارهای هم ریخت و تعدد نمودار ها ممکن است کمی از ملموس بودن آن ها از نظر توسعه دهندگان بکاهد.

متدولوژی Booch : در این متدولوژی همانطور که ذکر شد، محصولات ساده و قابل فهم هستند، بنابراین از دید کاربر ملموس هستند. همچنین محصولاتی که تولید می شوند در راستای جلو بردن پروژه و اضافه کردن کلاس‌ها و اشیاء و تعاملات آن ها و مدلسازی ساختاری و رفتاری هستند. بنابراین مفید هستند و درنتیجه برای توسعه دهندگان نیز ملموس هستند.

متدولوژی OMT : در این متدولوژی ۳ مدل وجود دارد که در هر یک نمودار هایی رسم می شوند. در مدل شیء class diagram و data dictionary، در مدل پویا event trace diagram و state transition diagram و در مدل وظیفه ای نمودار DFD ترسیم می شود. این نمودار ها خیلی پیچیدگی ندارند و برای کاربر قابل فهم می‌باشند. همچنین در هر مرحله مدل ها و نمودار ها با هدف پیشرفت پروژه تکمیل می شوند و مفید هستند. اما وجود نمودار DFD خیلی توجیهی ندارد بنابراین می تواند کمی متدولوژی را از دید توسعه دهندگان ناملموس کند.

- قابل ردیابی به نیازمندی ها

محصولات باید به عنوان تحقق مستقیم یا غیرمستقیم نیازمندی ها، یا از طریق استفاده از سناریوهای ارزیابی مبتنی بر نیازمندی ها قابل ردیابی باشند.

متدولوژی UP : در این متدولوژی نیازمندی در use case ها مدل می شوند و این متدولوژی use-case-driven است، یعنی همه کارها بر مبنای use case ها انجام می شوند. بنابراین نیازمندی ها قابل ردیابی مستقیم بوسیله use case ها هستند.

متدولوژی Booch : در این متدولوژی همانطور که ذکر شد، با اینکه استخراج نیازمندی ها به عنوان یک فاز جداگانه در macro process در نظر گرفته شده است و مدلسازی قلمرو مسئله با استفاده از نیازمندی ها صورت می گیرد، نیازمندی‌ها بطور مستقل مدل نمی شوند و همچنین سناریو های کاربرد ایجاد نمی شوند تا بتوان از روی آن ها نیازمندی ها را ردیابی کرد. بنابراین این متدولوژی قابلیت ردیابی به نیازمندی ها را ندارد.

متدولوژی OMT : این متدولوژی با اینکه در فاز تحلیل نیازمندی ها را استخراج می کند، اما فعالیتی با عنوان تحلیل نیازمندی ها ندارد و نمی توان گفت کاملا مبتنی بر نیازمندی است. همچنین در مدل ها و نمودار ها چیزی که بتواند ما را به نیازمندی ها برساند وجود ندارد و ردیابی به نیازمندی ها از طریق DFD مشکل است. در نتیجه می توان گفت که این متدولوژی قابلیت ردیابی به نیازمندی ها را ندارد.

مقایسه متدولوژی UP و Booch :

شباهت ها:

هر دو متدولوژی تا حدی ملموس هستند.

تفاوت ها:

متدولوژی Booch بدلیل تعداد محصولات کمتر و قابل فهم بودن آن ها تا حدی آزمون پذیر است در حالی که متدولوژی UP به دلیل تعداد بالای محصولات و وجود وابستگی بین آن ها آزمون پذیر نیست. متدولوژی UP به دلیل use-case-driven بودن، ردیابی به نیازمندی ها را بطور مستقیم دارد در حالی که در متدولوژی Booch ردیابی به نیازمندی ها مشکل است. در متدولوژی UP ، تعدد مدل ها ممکن است کمی به ملموس بودن خدشه وارد کند.

مقایسه:

بنابراین متدولوژی UP از نظر ردیابی به نیازمندی ها نسبت به متدولوژی Booch برتری دارد. متدولوژی Booch نیز از نظر آزمون پذیری و کمی هم از نظر ملموس بودن نسبت به متدولوژی UP برتری دارد.

مقایسه متدولوژی UP و OMT :

شباهت ها:

در هر دو متدولوژی محصولات قابل فهم هستند و پیچیدگی کمی دارند. هر دو متدولوژی تا حدی ملموس هستند. هر دو متدولوژی تا حدی از نظر کاربران ملموس هستند و ملموس بودن آنان از نظر توسعه دهندگان کمی دچار مشکل است. (در متدولوژی UP به دلیل تعدد مدل و وجود وابستگی و در متدولوژی OMT به دلیل وجود نمودار DFD)

تفاوت ها:

متدولوژی OMT به دلیل ساده بودن و تعداد کم محصولات آزمون پذیر است در حالی که در متدولوژی UP تعداد مدل ها و وابستگی بینشان زیاد است که باعث می شود آزمون پذیر نباشد. در متدولوژی OMT با اینکه مدل ها در ابتدا بر مبنا نیازمندی ها ساخته می شوند اما راهکاری برای اینکه بعداً بتوان به نیازمندی ها رسید ندارد و در نتیجه ردیابی به نیازمندی ها را ندارد. در حالی که متدولوژی UP ، requirements-driven است و ردیابی به نیازمندی ها بصورت مستقیم وجود دارد.

مقایسه:

متدولوژی UP از لحاظ ردیابی به نیازمندی ها نسبت به متدولوژی OMT برتری دارد. و متدولوژی OMT از لحاظ آزمون پذیر بودن نسبت به متدولوژی UP برتری دارد.

۴-۱-۷- تشویق مشارکت فعال کاربران

بررسی اینکه آیا متدولوژی کاربران را به حضور فعال در فرایند ایجاد نرم افزار تشویق می کند یا خیر، برای مدیریت ریسک و تضمین کیفیت ضروری است. برنامه ریزی و جلسات مرور[۹۵] با کاربران مفید می باشد.

متدولوژی UP : در این متدولوژی در هر فاز محصولاتی که تولید می شوند به ذی نفعان نشان داده می شود و تایید آن ها گرفته می شود. برای مثال در فاز inception، هدف پروژه، نیازمندی های اساسی و حوزه سیستم ، تخمین هزینه و زمان، مورد توافق ذی نفعان قرار می گیرد. در فاز elaboration، در مورد طرح توجیه اقتصادی و project plan با ذی نفعان به توافق می رسند. در فاز construction محصول نهایی مورد توافق ذی نفعان قرار می‌گیرد. و در نهایت در فاز transition تایید کاربران برای محصولی که در محیط آن ها نصب شده است، گرفته

[۹۵] session review

می شود. بنابراین کاربران در این متدولوژی مشارکت فعال دارند. اما اینگونه نیست که عضوی از کاربران عضو تیم باشد.

متدولوژی Booch : در این متدولوژی، macro process به عنوان یک چارچوب مدیریتی برای micro process عمل می‌کند و فعالیت هایی مثل تضمین کیفیت، مرور کد و مستندات در این بخش انجام می شوند. همچنین در macro process ، به مشتری و نیازهایش مثل کیفیت، کامل بودن و زمانبندی، توجه می شود. و این کارها به صورت تکراری انجام می پذیرند. بنابراین با اینکه در این متدولوژی اشاره ای به برگزاری جلسات با کاربران و حضور فعال کاربر نشده است، اما برای تحقق فعالیت های مدیریتی ذکر شده در بالا، احتمالا تعامل با کاربر وجود دارد.

متدولوژی OMT : در این متدولوژی تنها در فاز اول یعنی تحلیل، نیازمندی های کاربر استخراج می شود و از آن پس تعامل و مشارکت فعال با کاربر صورت نمی گیرد.

مقایسه متدولوژی UP و Booch :

شباهت ها:

در هر دو متدولوژی به نظر کاربران اهمیت داده می شود اما در هیچ یک از آن ها کاربر به عنوان عضو تیم در نظر گرفته نشده است.

تفاوت ها:

در متدولوژی UP اشاره شده است که در اخر هر فاز محصولاتی به کاربر نشان داده می شوند و تایید آن ها گرفته می شود اما در متدولوژی Booch بصورت دقیق بیان نشده است که چگونه نظر و تایید کاربر گرفته می‌شود. اما ذکر شده است که تمرکز فعالیت های macro process روی کیفیت مد نظر کاربران است.

مقایسه:

در متدولوژی UP حضور فعال کاربران پررنگ تر است و یک مزیت نسبت به متدولوژی Booch محسوب می‌شود.

مقایسه متدولوژی UP و OMT :

شباهت ها:

در هیچ یک از دو متدولوژی کاربر به عنوان عضو تیم قرار ندارد.

تفاوت ها:

در متدولوژی UP دائما نظر کاربر پرسیده می شود و از او تایید گرفته می شود اما در متدولوژی OMT جز در ابتدا هنگام استخراج نیازمندی ها، دیگر ارتباط با کاربر صورت نمی گیرد.

مقایسه:

در متدولوژی UP کاربر حضور فعال دارد در حالی که در متدولوژی OMT اینگونه نیست. بنابراین این برتری متدولوژی UP نسبت به متدولوژی OMT می باشد.

۴-۱-۸- قابلیت اجرا[۹۶] و قابلیت اجرا بصورت کارا[۹۷]

قابلیت اجرا یعنی بتوان یک فرایند را اجرا کرد. می تواند به پیچیدگی آن فرایند یا ماهیت پروژه وابسته باشد.

قابلیت اجرا به صورت کارا یعنی فرایند استفاده مؤثر داشته باشد. موارد زیر تاثیرگذار هستند:

- پیچیدگی واحدهای فرایند
- وظایفی که توسعه دهندگان را از فعالیت های جریان اصلی منحرف می کند یا آن ها را به جزئیات غیرضروری مشغول می کند. می توان با استفاده از تکنیک هایی مانند جلسات مدیریت تیم، مدل های مبتنی بر نیازمندی و معماری سیستم، توسعه دهندگان را متمرکز کرد.
- وابستگی به تکنیک ها و استراتژی های مستعد خطا
- وابستگی به ابزار و فناوری های خاص
- فقدان استراتژی مدیریت پروژه کافی

متدولوژی UP : این متدولوژی فرایند پیچیده ای دارد. همانطور که گفته شد از ۴ فاز inception، elaboration، construction و transition تشکیل شده است که در هرکدام از این فاز ها iteration های نیازمندی‌ها، تحلیل، طراحی، اجرا و تست، انجام می شوند. درست است که ماهیت تکراری-افزایشی آن خیلی مزایا دارد اما پیچیدگی فازها را بیشتر هم کرده است. بنابراین کمی به قابلیت اجرا خدشه وارد می شود. برای استفاده از UP باید آن را متناسب با پروژه شخصی سازی کنیم. البته شایان ذکر است که پیچیدگی فرایند متدولوژی UP کمتر از متدولوژی RUP است و اجرا کردن آن ساده تر از RUP است.

همانطور که گفتیم چون قابل اجرا بودن متدولوژی دچار خدشه است بنابراین صحبت از قابل اجرا بودن بصورت کارا مطرح نیست، اما نکات مثبت و منفی وجود دارد که در ادامه به آن ها می پردازیم. این متدولوژی مبتنی بر

[۹۶] practicability
[۹۷] practicality

نیازمندی هاست و همچنین معماری مبنا قابل اجرا دارد که مبنای کارها قرار می گیرد، بنابراین این تکنیک ها توسعه دهندگان را متمرکز نگه می دارند. وابستگی به تکنیک ها و استراتژی های مستعد خطا وجود ندارد و همچنین استراتژی های مدیریت پروژه (جزئیات در بخش ۳-۱-۴) خیلی خوب انجام می شوند. اما ضعفی که وجود دارد تأکید آن روی استفاده از UML است که در بعضی موارد دچار مشکل است و کافی نیست.

متدولوژی Booch : این متدولوژی یک متدولوژی تکراری-افزایشی است که در آن macro process به عنوان یک چارچوب کنترلی برای micro process تعریف شده است. macro process از ۵ فاز تشکیل شده است. در هر فاز micro process بصورت تکراری انجام می شود. ۵ فاز آن عبارتند از مفهوم‌سازی، تحلیل، طراحی، تکامل و نگهداری. خود micro process از ۴ مرحله تشکیل شده است که بصورت تکراری انجام می‌شوند. بنابراین این متدولوژی فرایند پیچیده ای ندارد و اتفاقا ماهیت تکراری-افزایشی آن را ساده تر نیز کرده است. در نتیجه Booch قابل اجراست.

در این متدولوژی همانطور که ذکر شد، واحدهای فرایند پیچیدگی ندارند. نمودار هایی که در هر فاز macro process و در هر مرحله micro process ایجاد می شوند، در راستای پیشبرد اهداف هستند و توسعه دهندگان را به کارهای غیرضروری مشغول نمی کنند. این متدولوژی به تکنیک ها و استراتژی های مستعد خطا وابسته نیست و وابستگی به ابزار خاصی هم ندارد. همچنین در این متدولوژی macro process مثل یک چارچوب مدیریتی عمل می کند و پیشبرد کار ها طوری که کیفیت تضمین شود را بررسی می کند. بنابراین این متدولوژی قابلیت اجرا بصورت کارا را دارد.

متدولوژی OMT : این متدولوژی ۳ مدل شیء، پویا و وظیفه ای دارد که طی ۵ فاز تحلیل، طراحی سیستم، طراحی شیء، کد و تست، تکمیل می شوند. در فاز تحلیل هدف این است که یک مدل صحیح از دنیای واقع بسازیم، در همین جهت این سه مدل به طور تکراری ساخته می‌شوند. در فاز طراحی سیستم، ساختار سطح بالای سیستم و ویژگی های قلمرو جواب مشخص می شود. در فاز طراحی شیء، کلاس‌های قلمرو جواب و روابط بینشان به سه مدلی که ساختیم اضافه می‌شود. در فاز کد و تست هم که پیاده سازی و آزمون نرم افزار انجام می شود. بنابراین فرایند خیلی پیچیدگی ندارند. بنابراین این متدولوژی قابل اجرا است.

متدولوژی OMT همانطور که ذکر شد از ۳ مدل تشکیل شده است که طی ۵ فاز تکمیل می شوند و این ۵ فاز همانطور که ذکر شد، ساده و فاقد پیچیدگی هستند. همچنین فاقد تکنیک های مستعد خطا و وابستگی به ابزار و فناوری خاص می باشد. اما در این متدولوژی در مدل وظیفه ای نمودار DFD ترسیم می شود که چون شیء گرا نیست خیلی کاربرد ندارد و بنابراین توسعه دهندگان را به کار غیر ضروری مشغول می کند. و اینکه به استراتژیی در راستای مدیریت پروژه در OMT اشاره نشده است.

مقایسه متدولوژی UP و Booch :

شباهت ها:

در هر دو متدولوژی به ساخت معماری سیستم که از تکنیک های حفظ تمرکز توسعه دهندگان است، استفاده می شود. هیچ یک از این دو وابسته به تکنیک های خطا خیز نیستند و هر دو متدولوژی از استراتژی های مدیریت پروژه استفاده می کنند.

تفاوت ها:

فرایند متدولوژی UP پیچیده است که این قابل اجرا بودن آن را مشکل می کند.

مقایسه:

متدولوژی Booch فرایند ساده ای دارد که این باعث می شود قابل اجرا باشد و همچنین با توجه به مواردی که ذکر شد قابل اجرا بصورت کارا نیز هست. اما متدولوژی UP به دلیل پیچیده بودن فرایند آن قابل اجرا بودن آن دچار مشکل است و بنابراین قابلیت اجرا بصورت کارا نیز برای آن مطرح نیست. بنابراین این برتری متدولوژی Booch نسبت به متدولوژی UP است.

مقایسه متدولوژی UP و OMT :

شباهت ها:

در هر دو متدولوژی به ساخت معماری سیستم که از تکنیک های حفظ تمرکز توسعه دهندگان است، استفاده می شود و هیچ یک از این دو وابسته به تکنیک های خطا خیز نیستند.

تفاوت ها:

فرایند متدولوژی UP پیچیده است که این قابل اجرا بودن آن را مشکل می کند. در متدولوژی OMT در مدل وظیفه ای نمودار DFD ترسیم می شود که چون شیء گرا نیست خیلی کاربرد ندارد و بنابراین توسعه دهندگان را به کار غیر ضروری مشغول می کند. و اینکه به استراتژیی در راستای مدیریت پروژه در OMT اشاره نشده است. در حالی که در متدولوژی UP مدیریت پروژه به خوبی انجام می شود و توسعه دهندگان نیز به کارهای غیر ضروری مشغول نمی شوند.

مقایسه:

متدولوژی OMT به دلیل ساده بودن فرایند آن قابل اجرا است در حالی که فرایند متدولوژی UP پیچیده است که باعث می شود قابل اجرا بودن آن مشکل باشد. همچنین متدولوژی OMT تا حدی قابل اجرا بصورت کارا است در حالی که متدولوژی UP به دلیل قابل اجرا نبودن، قابل اجرا بودن بصورت کارا برای آن مطرح نیست. بنابراین متدولوژی OMT در این معیار نسبت به متدولوژی UP برتری دارد.

۴-۱-۹- مدیریت پیچیدگی

پیچیدگی واحد های کاری باید قابل مدیریت باشند. تکنیک هایی که برای این کار استفاده می شوند عبارتند از:

- Partitioning : شکستن کار ها به قطعات کوچکتر در یک سطح انتزاع
- Layering : شکستن کار ها به قطعات کوچکتر در سطوح مختلف انتزاع

متدولوژی UP : همانطور که گفته شد این متدولوژی از ۴ فاز inception، elaboration، construction و transition تشکیل شده است که در هرکدام از این فاز ها iteration های نیازمندی‌ها، تحلیل، طراحی، اجرا و تست، انجام می شوند. وجود فازهای مختلف در سطوح انتزاع مختلف را می توان نوعی layering دانست. همچنین در هر فاز در iteration ها فعالیت ها خود در ۵ جریان کاری انجام می شوند که نوعی partitioning است. بنابراین این متدولوژی بدلیل ماهیت تکراری-افزایشی و اینکه کارها در بخش های کوچک انجام می شوند، از مدیریت پیچیدگی خوبی برخوردار است.

متدولوژی Booch : فرایند این متدولوژی تکراری-افزایشی است، بنابراین مدیریت پیچیدگی را در ذات خود دارد چرا که فعالیت ها در دو فرایند macro process و micro process انجام می شوند و در هر فاز macro process ، مراحل micro process بصورت تکراری اجرا می شوند. بنابراین نه تنها که کارها در سطوح انتزاع مختلف در macro process شکسته شده اند، در هر سطح انتزاع کار ها در micro process نیز شکسته می شوند. در نتیجه این متدولوژی از تکنیک های layering و partitioning استفاده می کند و مدیریت پیچیدگی را انجام می دهد.

متدولوژی OMT : این متدولوژی از ۵ فاز تحلیل، طراحی سیستم، طراحی شیء، کد و تست تشکیل شده است. در فاز تحلیل ۳ مدل شیء، پویا و وظیفه ای می سازد. در فاز طراحی سیستم ساختار سطح بالای سیستم را مشخص می کند. و در فاز طراحی شیء به طراحی تفضیلی می پردازد؛ یعنی جزئیات را به مدل هایی که داشتیم

اضافه می کند. بنابراین این متدولوژی کار ها را در سطوح مختلف انتزاع می شکند و انجام می دهد. می توان گفت که از مدیریت پیچیدگی خوبی برخوردار است.

مقایسه متدولوژی UP و Booch :

شباهت ها:

هر دو متدولوژی فرایند تکراری-افزایشی دارند و کار ها را در سطوح انتزاع مختلف و در هر سطح انتزاع نیز می‌شکنند. بنابراین هر دو از تکنیک های partitioning و layering استفاده می کنند که باعث می شود هر دو از مدیریت پیچیدگی خوبی برخوردار باشند.

تفاوت ها: -

مقایسه: –

مقایسه متدولوژی UP و OMT :

شباهت ها:

هر دو متدولوژی کارها را به کارهای کوچکتر می شکنند بنابراین مدیریت پیچیدگی را انجام می دهند.

تفاوت ها:

فرایند متدولوژی UP تکراری-افزایشی است که این باعث می شود کار ها هم در سطوح انتزاع مختلف و هم در یک سطح انتزاع شکسته شوند بنابراین بهتر از متدولوژی OMT به مدیریت پیچیدگی می پردازد.

مقایسه:

متدولوژی UP به دلیل فرایند تکراری-افزایشی، بهتر از متدولوژی OMT مدیریت پیچیدگی را انجام می دهد.

۴-۱-۱۰- توسعه پذیری / مقیاس پذیری / پیکربندی / انعطاف پذیری

- توسعه پذیری

فرایند باید یک هسته توسعه پذیر باشد که نقاط گسترش و مکانیسم های آن به صراحت مشخص شده باشد.

متدولوژی UP : این متدولوژی یک هسته ی توسعه پذیر نیست و نقاط گسترش برای آن مشخص نشده است. بنابراین این متدولوژی توسعه پذیر نیست.

متدولوژی Booch : در این متدولوژی با اینکه کار ها به تدریج انجام و تکمیل می شوند، فرایند یک هسته ای با نقاط قابل گسترش مشخص نمی باشد. در نتیجه این متدولوژی توسعه پذیر نیست.

متدولوژی OMT : در این متدولوژی با اینکه کار ها به تدریج انجام و تکمیل می شوند، فرایند یک هسته ای با نقاط قابل گسترش مشخص نمی باشد. در نتیجه این متدولوژی توسعه پذیر نیست.

- مقیاس پذیری

فرایند باید برای پروژه هایی با اندازه ها و سطوح بحران[۹۸] مختلف قابل اجرا باشد.

متدولوژی UP : این متدولوژی همانطور که ذکر شد یک فرایند تکراری-افزایشی دارد. همچنین از نکات مثبت دیگری نیز برخوردار است که در نتیجه آن ها از مدیریت ریسک، مدیریت پروژه و تضمین کیفیت خوبی برخوردار است. از دیگر مزایا می توان به زبان مدلسازی آن اشاره کرد که زبان غنی UML است و همچنین می تواند سطوحی از فرمالیزم را ارائه دهد. البته باید اشاره کرد که متدولوژی UP فازی برای نگهداری در نظر نگرفته است و در نگهداری دارای ضعف است.

بنابراین با توجه به داشتن فرایند سنگین و با جزئیات، مدیریت پروژه و مدلسازی قوی، می توان از متدولوژی UP در پروژه های بزرگ با دخالت تیم های متعدد استفاده کرد. همچنین بدلیل برخوردار بود از سطوحی از فرمالیزم می توان از آن در پروژه ها با سطح بحرانیت بالا نیز استفاده کرد. در نتیجه متدولوژی UP مقیاس پذیر است. البته بسته به نوع پروژه باید شخصی سازی شود.

متدولوژی Booch : این متدولوژی یک متدولوژی تکراری-افزایشی است در نتیجه کار ها به صورت تکراری و تدریجی انجام می شوند. همچنین macro process به عنوان یک چارچوب مدیریتی برای micro process عمل می کند و کیفیت را نیز نضمین می کند. بنابراین می توان پروژه هایی با سایز های بزرگ را نیز انجام داد. پروژه ها با سطوح مختلف بحرانیت نیز قابل اجرا با این متدولوژی هستند اما Booch به علت نداشتن مدلسازی فرمال، برای پروژه ها با سطوح بحرانیت بالا مناسب نیست.

متدولوژی OMT : این متدولوژی دارای ۵ فاز است که در فاز دوم یعنی طراحی سیستم، سیستم را به زیر سیستم‌ها می شکند و زیر سیستم ها را به کارها و پردازنده ها اختصاص می دهد. در نتیجه برای پروژه های

[۹۸] life critical - essential money critical - discretionary money critical - comfort critical

بزرگتر می تواند زیر سیستم های بیشتری تعریف کند و به آن ها بپردازد. اما به دلیل استفاده نکردن از تکنیک‌های مدیریت پروژه نمی توان از آن برای پروژه با سایز های بالا استفاده کرد.

اما به دلیل نداشتن مکانیزم هایی برای مدیریت ریسک، نداشتن مدلسازی فرمال و همچنین نداشتن مرحله نگهداری، برای پروژه ها با سطح بحرانیت بالا و پروژه هایی که نیاز جدی به مرحله نگهداری دارند، مناسب نیست.

- پیکربندی

بتوان در شروع پروژه فرایند را به منظور تناسب آن با وضعیت پروژه پیکربندی کرد.

متدولوژی UP : این متدولوژی نسخه ساده شده RUP است، از این نظر قابلیت پیکربندی اولیه را دارد.

متدولوژی Booch : در این متدولوژی اشاره ای به پیکربندی اولیه مسئله با توجه به شرایط آن نشده است.

متدولوژی OMT : در این متدولوژی اشاره ای به پیکربندی اولیه مسئله نشده است و کار ها به طور ثابت با فاز تحلیل شروع می شوند و سه مدل شیء، پویا و وظیفه ای ساخته می شوند.

- انعطاف پذیری

فرآیند باید در حین اجرا قابل تنظیم باشد. تکنیک های جلسات تکراری مرور فرایند و تجدید نظرهای مبتنی بر بازخورد، مفید هستند.

متدولوژی UP : این متدولوژی فرایند تکراری-افزایشی دارد و در هر فاز project plan مورد بازنگری قرار می گیرد و اصلاح می شود. همچنین محصولات دیگر نیز به کاربران نشان داده می شوند و کاربران بازخورد می دهند. بنابراین فرایند در حین اجرا قابل تنظیم است و می توان گفت متدولوژی UP انعطاف‌پذیر است.

متدولوژی Booch : در این متدولوژی همانطور که ذکر شد، مراحل micro process در هر فاز macro process بصورت تکراری انجام می شوند و macro process کنترل مدیریتی روی آن ها انجام می دهد. برای مثال می‌توان به مرور کد اشاره کرد. بنابراین فرایند می تواند در حین اجرا قابل تنظیم باشد اما چون اشاره‌ای به جلسات مرور فرایند نشده است نمی توان کاملا آن را انعطاف پذیر دانست.

متدولوژی OMT : این متدولوژی جلسات تکراری مرور فرایند تشکیل نمی دهد و تجدید نظر های مبتنی بر بازخورد ندارد چرا که با کاربر جز در ابتدای کار، تعامل ندارد. در نتیجه انعطاف پذیری ندارد.

مقایسه متدولوژی UP و Booch :

شباهت ها:

هیچ یک از دو متدولوژی توسعه پذیر نیستند. هر دو متدولوژی تا حدی مقیاس پذیر و انعطاف پذیر هستند.

تفاوت ها:

متدولوژی UP قابلیت پیکربندی اولیه دارد در حالی که متدولوژی Booch ندارد. متدولوژی UP می تواند سطوحی از فرمالیزم داشته باشد که باعث می شود برای پروژه ها با سطوح بحرانیت بالا نیز مناسب باشد در حالی که متدولوژی Booch مدلسازی فرمال ندارد و برای پروژه ها با سطح بحرانیت بالا مناسب نیست. همچنین فرایند متدولوژی UP سنگین تر است و جزئیات بیشتری دارد بنابراین بهتر می توان از آن برای پروژه ها با سایز بالا استفاده کرد.

مقایسه:

متدولوژی UP قابلیت پیکربندی اولیه دارد و نسبت به متدولوژی Booch مقیاس پذیرتر است، بنابراین از این نظر ها نسبت به متدولوژی Booch برتری دارد.

مقایسه متدولوژی UP و OMT :

شباهت ها:

هیچ یک از دو متدولوژی توسعه پذیر نیستند.

تفاوت ها:

متدولوژی UP، متدولوژی سنگینی است و فرایند آن تکراری-افزایشی است و خیلی جزئیات دارد، فعالیت های چتری مدیریت ریسک، مدیریت پروژه و تضمین کیفیت را به خوبی پوشش می دهد، مدلسازی آن قوی است و از فرمالیزم پشتیبانی می کند. بنابراین متدولوژی UP مقیاس پذیر است در حالی که متدولوژی OMT هیچ یک از موارد گفته شده را ندارد.

متدولوژی UP را می توان پیکربندی اولیه کرد و همچنین بدلیل فرایند تکراری-افزایشی آن می توان فرایند را بازنگری کرد و این متدولوژی انعطاف پذیر نیز هست.

مقایسه:

متدولوژی UP از نظر داشتن قابلیت های مقیاس پذیری، پیکربندی اولیه و انعطاف پذیری نسبت به متدولوژی OMT برتری دارد.

۴-۱-۱۱- حوزه کاربرد[۹۹]

حوزه کاربرد فرایند به زمینه ای که در آن استفاده می شود، وابسته است؛ اما سیستم های اطلاعاتی حتما باید پوشش داده شوند.

متدولوژی UP : این متدولوژی همانطور که ذکر شد یک فرایند تکراری-افزایشی دارد که بسیار به مدیریت ریسک کمک می کند. نکات مثبت دیگری که می توان به آن اشاره کرد عبارتند از اینکه تحلیل امکان سنجی انجام می‌دهد، prototype های دورریختنی برای پیاده سازی use case های ریسکی و همچنین برای انتخاب معماری می‌سازد، در هر فاز برای فاز بعدی برنامه ریزی می کند، در هر فاز تأیید کاربر را برای محصولات ساخته شده می‌گیرد، معیارهایی برای تضمین کیفیت دارد و در جریان های کاری که در هر فاز تکرار می شوند، شامل مرحله تست است. همه این ها کمک می کنند که این متدولوژی از مدیریت ریسک، مدیریت پروژه و تضمین کیفیت خوبی برخوردار باشد. از جهتی requirement-driven هم می باشد. از دیگر مزایا می توان به زبان مدلسازی آن اشاره کرد که UML است و همانطور که می دانیم زبان مدلسازی غنی می باشد و همچنین می تواند سطوحی از فرمالیزم را ارائه دهد. البته باید اشاره کرد که متدولوژی UP فازی برای نگهداری در نظر نگرفته است و برای نگهداری همان تکرار فاز ها و جریان های کاری موجود در هر فاز را پیشنهاد داده است؛ اما می دانیم که نمی توان از فعالیت های تکراری برای نگهداری استفاده کرد. بنابراین این متدولوژی در نگهداری دارای ضعف است.

بنابراین با تمام این تفاسیر می توان گفت که این متدولوژی را برای اکثر سیستم ها، از جمله سیستم های اطلاعاتی، می توان بکار برد. مخصوصا بدلیل برخوردار بود از سطوحی از فرمالیزم می توان از آن در پروژه ها با سطح بحرانیت بالا نیز استفاده کرد.

متدولوژی Booch : این متدولوژی همانطور که در بخش قبل ذکر شد، دارای مقیاس پذیری در پروژه ها با سایزهای مختلف می باشد. و خاصیت تکراری-افزایشی آن یک مزیت بزرگ محسوب می شود. همچنین چرخه عمر عمومی نرم‌افزار را تا حد خوبی (بجز تحلیل امکانپذیری در فاز تعریف) پوشش می دهد. بنابراین می توان از Booch در اکثر پروژه‌ها استفاده کرد با این استثناء که به دلیل نداشتن مدلسازی فرمال، برای پروژه ها با سطح بحرانیت بالا مناسب نیست.

[۹۹] application scope

متدولوژی OMT : این متدولوژی به دلیل دارا بودن سه مدل شیء (ساختاری)، پویا (رفتاری) و وظیفه ای، وجوه اصلی مورد نیاز پروژه ها را دارا هست و همچنین مراحل تحلیل، طراحی معماری، طراحی تفضیلی، پیاده سازی و تست را به خوبی انجام می دهد و می توان از آن در اکثر پروژه ها استفاده کرد (برای مثال در ناسا هم استفاده شده است). اما همانطور که گفته شد بدلیل نداشتن مکانیزم هایی برای مدیریت ریسک و مدیریت پروژه، نداشتن مدلسازی فرمال و همچنین نداشتن مرحله ی نگهداری، برای پروژه ها با سایز و سطح بحرانیت بالا و پروژه هایی که نیاز جدی به مرحله نگهداری دارند، مناسب نیست.

مقایسه متدولوژی UP و Booch :

شباهت ها:

هر دو متدولوژی برای اکثر سیستم ها مخصوصا سیستم های اطلاعاتی قابل استفاده هستند.

تفاوت ها:

متدولوژی Booch را به علت نداشتن مدلسازی فرمال نمی توان برای پروژه ها با سطح بحرانیت بالا استفاده کرد در حالی که از متدولوژی UP می توان در پروژه ها با سطح بحرانیت بالا نیز استفاده کرد. همچنین متدولوژی UP سنگین تر است و برای پروژه ها با سایز بالا مناسب تر است.

مقایسه:

حوزه کاربرد متدولوژی UP وسیع تر است برای سیستم های بزرگ و بحرانی نیز مناسب است و این یک برتری نسبت به متدولوژی Booch محسوب می شود.

مقایسه متدولوژی UP و OMT :

شباهت ها:

هر دو متدولوژی برای اکثر سیستم ها مخصوصا سیستم های اطلاعاتی قابل استفاده هستند.

تفاوت ها:

متدولوژی UP فعالیت های چتری مدیریت ریسک، مدیریت پروژه و تضمین کیفیت را داراست در حالی که متدولوژی OMT آن ها را پوشش نمی دهد. فرایند متدولوژی UP تکراری-افزایشی است و حاوی جزئیات بالاست در حالی که فرایند متدولوژی OMT ترتیبی و ساده تر است. همچنین متدولوژی UP دارای مدلسازی فرمال نیز

هست در حالی که متدولوژی OMT مدلسازی فرمال ندارد. بنابراین متدولوژی UP برای پروژه ها با سایز بالا و سطوح بحرانیت بالا نیز مناسب است.

مقایسه:

با توجه به موارد بالا، حوزه کاربرد متدولوژی UP وسیع تر از متدولوژی OMT است و نسبت به آن برتری دارد.

۴-۲- معیار های زبان مدلسازی

۴-۲-۱- پشتیبانی از مدلسازی شی گراء سازگار، دقیق و بدون ابهام

- زبان مدلسازی باید دیدگاه های مدل سازی متنوع ساختاری، وظیفه ای و رفتاری را شامل شود.
- مدل‌سازی منطقی تا فیزیکی داشته باشد (دامنه فرآیند کسب‌وکار/ قلمرو مسئله به قلمرو جواب به قلمرو اجرا).
- بتوانیم در سطوح مختلف درشت دانگی و انتزاع، مدلسازی کنیم (سازمان - سیستم – زیر سیستم/پکیج – inter-object – intra-object).
- وجوه مدلسازی formal و informal را داشته باشد.

متدولوژی UP : زبان مدلسازی این متدولوژی، UML است.

- ✓ زبان مدلسازی UML هر سه وجه ساختاری، وظیفه ای و رفتاری را دارا می باشد. برای مثال مدلسازی ساختاری آن شامل class diagram ، object diagram ، package diagram و component diagram ، مدلسازی رفتاری آن شامل activity diagram ، sequence diagram ، state transition diagram و communication diagram و مدلسازی وظیفه‌ای آن شامل use case diagram می باشد.
- ✓ مدلسازی از منطقی تا فیزیکی را شامل می شود. برای مثال در تحلیل class diagram برای کلاس های قلمرو مسئله ایجاد می شود و در طراحی جزئیات قلمرو جواب به آن ها اضافه می شود و ممکن است کلاس های جدید قلمرو جواب هم به آن اضافه شود. و سپس همین کلاس ها پیاده سازی می شوند. یا مثلا در تحلیل sequence diagram تعامل بین اشیاء قلمرو مسئله را برای تحقق use case ها نشان می دهد و در طراحی کلاس ها و اشیاء قلمرو جواب نیز به آن اضافه می شود و نحوه تعامل آن ها نیز برای تحقق use case ها نمایش داده می شود. از دیگر مدل های قلمرو مسئله(مدل های منطقی) می توان به object diagram ، package diagram ، activity diagram و state transition diagram اشاره کرد. از دیگر مدل های قلمرو جواب (مدل های فیزیکی) می توان به object diagram طراحی، component

deployment diagram ، diagram و state transition diagram برای کلاس های طراحی اشاره کرد. پیاده سازی هم در واقع نوع دیگری از مدلسازی فیزیکی است که قلمرو جواب را مدل می کند.

- ✓ مدلسازی در سطوح مختلف انتزاع به جز در سطح سازمان، انجام می شود.
- ✓ مدلسازی فرمال در UML با استفاده از OCL انجام می شود.

متدولوژی Booch : این متدولوژی از زبان مدلسازی خاص خود استفاده می کند.

- ✓ در micro process که فعالیت های ریزدانه ایجاد نرم افزار انجام می شوند محصولاتی تولید می شوند. از جمله این محصولات نمودار های حالت و نمودار های تعامل هستند که مدلسازی رفتاری را انجام می‌دهند، نمودار های شیء، نمودار های کلاس و نمودار های ماژول هستند که مدلسازی ساختاری را نشان می‌دهند.
- ✓ در macro process که شامل ۵ فاز ترتیبی است، ابتدا نیازمندی ها استخراج می شوند و در فاز تحلیل قلمرو مسئله مدل می شود(مدلسازی منطقی). سپس در فاز طراحی معماری سیستم ایجاد می شود و جزئیات قلمرو جواب مشخص می شود و به مدل ها اضافه می شود، بنابراین قلمرو جواب نیز مدل می‌شود(مدلسازی فیزیکی). در فاز تکامل، طراحی تفضیلی کامل می شود و پیاده سازی صورت می گیرد، بنابراین قلمرو اجرا نیز مدلسازی می شود(مدلسازی فیزیکی).
- ✓ همانطور که ذکر شد در مراحل micro process نمودارهایی بسته به سطح انتزاعی که هستیم تولید یا تکمیل می‌شوند. برای مثال نمودار های کلاس در سطوح انتزاع بالاتر می توانند مدلسازی در سطح سیستم را نشان دهند، نمودار های ماژول، می توانند مدلسازی در سطح زیر سیستم ها نشان دهند و نمودار های شیء و تعامل، مدلسازی در سطح inter-object را نشان می دهند.
- ✓ این متدولوژی مدلسازی فرمال ندارد. .

متدولوژی OMT :

- ✓ زبان مدلسازی این متدولوژی حاوی ۳ مدل شیء، پویا و وظیفه ای است که به ترتیب دیدگاه های ساختاری، رفتاری و وظیفه ای را پوشش می دهد.
- ✓ این متدولوژی ۵ فاز دارد که در فاز اول یعنی تحلیل، با استفاده از نیازمندی های کاربران، توسعه دهندگان و مدیران، قلمرو مسئله را مدل می کند. سپس مدل های شیء، پویا و وظیفه ای را در این قلمرو می‌سازد. در فاز بعدی یعنی طراحی سیستم، ساختار سطح بالای سیستم را مشخص می کند و وارد قلمرو جواب می شود. در فاز سوم یعنی طراحی شیء، به طراحی تفضیلی می پردازد و کلاس های قلمرو جواب را به مدل ها اضافه می کند. در آخر در فازهای کد و تست قلمرو اجرا را مدلسازی می کند. در نتیجه OMT مدلسازی از منطقی تا فیزیکی را دارد و برای مثال class diagram که در فاز تحلیل ساخته می شود

منطقی است و در طی انجام فاز ها وابسته به پیاده سازی خاص نرم افزاری می شود و جنبه فیزیکی به خود می گیرد.

- ✓ OMT سطوح مختلف انتزاع را مدلسازی می کند و در مورد سطوح مختلف درشت دانگی، در فاز تحلیل ۳ مدل شیء، پویا و وظیفه ای را می سازد که در مدل پویا state transition diagram و event-trace diagram ترسیم می کند که در ابتدا تعامل در سطح سیستم تعریف می شود و سپس داخل سیستم بررسی می شود که اشیاء چگونه با هم تعامل داشته باشند تا وظایف سیستمی را محقق کنند. بنابراین روابط درون و برون شیء ها را نمایش مشخص می شوند. همچنین در فاز طراحی سیستم، سیستم را به زیر سیستم ها می شکند و هر زیر سیستم مدل می شود.
- ✓ در OMT مدلسازی فرمال پوشش داده نمی شود.

مقایسه متدولوژی UP و Booch :

شباهت ها:

زبان مدلسازی هر دو متدولوژی وجوه ساختاری، رفتاری و وظیفه ای را شامل می شوند. مدلسازی از منطقی تا فیزیکی را دارند و مدلسازی در سطوح مختلف انتزاع به جز در سطح سازمان را انجام می دهند.

تفاوت ها:

زبان مدلسازی متدولوژی UP مدلسازی فرمال را با استفاده از OCL دارد در حالی که متدولوژی Booch مدلسازی فرمال را پوشش نمی دهد.

مقایسه:

زبان متدولوژی UP ، UML است که بسیار غنی است و دیدگاه های مختلف مدلسازی را بهتر از زبان مدلسازی Booch پوشش می دهد. همچنین دارای مدلسازی فرمال است که برتری نسبت به متدولوژی Booch محسوب می شود.

مقایسه متدولوژی UP و OMT :

شباهت ها:

زبان مدلسازی هر دو متدولوژی وجوه ساختاری، رفتاری و وظیفه ای را شامل می شوند. مدلسازی از منطقی تا فیزیکی را دارند و مدلسازی در سطوح مختلف انتزاع به جز در سطح سازمان را انجام می دهند.

تفاوت ها:

زبان مدلسازی متدولوژی OMT مدلسازی فرمال را پوشش نمی دهد در حالی که زبان مدلسازی متدولوژی UP که UML است، مدلسازی فرمال را با استفاده از OCL دارد.

مقایسه:

به دلیل اینکه زبان متدولوژی UP، UML است و همچنین فرمالیزم را نیز ساپورت می کند، از این نظر این متدولوژی نسبت به متدولوژی OMT برتری دارد.

۴-۲-۲- ارائه استراتژی‌ها و تکنیک‌ها برای مقابله با ناسازگاری مدل و مدیریت پیچیدگی مدل

اگر زبان های مدل سازی معناشناسی[۱۰۰] داشته باشند که محدودیت ها[۱۰۱] و وابستگی ها[۱۰۲] را تعریف کند، بررسی کردن ناسازگاری آسان تر خواهد بود.

زبان‌های مدل‌سازی باید شامل سازه‌هایی مانند پکیج های UML برای تسهیل مدیریت پیچیدگی باشند.

متدولوژی UP :

- ✓ همانطور که گفته شد زبان مدلسازی این متدولوژی UML است. تعداد مدل ها زیاد است و بین مدل ها وابستگی وجود دارد و برای مدیریت این ها هیچ تمهیدی اندیشیده نشده است. بنابراین زبان مدلسازی متدولوژی UP به مدیریت ناسازگاری نمی پردازد و برای مدیریت آن فرایند کمی کمک می کند.
- ✓ در این متدولوژی برای وجوه مختلف و سطوح انتزاع مختلف نمودارهای مختلفی وجود دارد. همچنین از پکیج‌های UML برای لایه بندی و قطعه بندی مدل ها استفاده می کند. بنابراین زبان مدلسازی متدولوژی UP به خوبی مدیریت پیچیدگی را انجام می دهد.

متدولوژی Booch :

- ✓ در این متدولوژی محصولاتی تولید می شوند که گاه وابستگی هایی نیز به یکدیگر خواهند داشت، اما زبان مدلسازی این متدولوژی تمهیداتی برای مدیریت ناسازگاری این محصولات در نظر نگرفته است و مدیریت ناسازگاری در فرایند انجام می شود.

---

[۱۰۰] semantic
[۱۰۱] constraints
[۱۰۲] dependencies

- ✓ همانطور که ذکر شد مدل ها در سطوح انتزاع مختلف ایجاد می شوند و همچنین وجوه ساختاری و رفتاری سیستم در نمودار های جداگانه نمایش داده می شود. بنابراین مدیریت پیچیدگی تا حدی در زبان مدلسازی هست اما از ساختار هایی برای مدیریت پیچیدگی استفاده نمی کند.

متدولوژی OMT :

- ✓ این متدولوژی مکانیزمی برای مدیریت ناسازگاری در زبان مدلسازی در نظر نگرفته است، اما به دلیل کم بودن مدل ها و ساده بودن آن ها، ناسازگاری کمتر اتفاق می افتد و درصورت بروز مدیریت آن ساده تر است.
- ✓ این متدولوژی از ۳ مدل شیء، پویا و وظیفه ای تشکیل شده است که در فاز تحلیل ساخته می شوند. مدل شیء ساختار ایستا سیستم را نشان می دهد، مدل پویا رفتار سیستم را با بررسی رفتار اشیاء در طول زمان و جریان کنترل و رویدادها در بین اشیاء نشان می دهد و مدل وظیفه ای فرایندهای داخلی سیستم را بدون توضیحات در مورد نحوه انجام این فرایندها توصیف می کند. سپس در فازهای بعدی این مدل ها کم کم تکمیل می شوند و جزئیات قلمرو جواب به آن ها اضافه می شود. بنابراین زبان مدلسازی OMT می تواند تا حدودی پیچیدگی را مدیریت کند، اما از سازوکار خاصی برای مدیریت پیچیدگی در زبان مدلسازی استفاده نشده است.

مقایسه متدولوژی UP و Booch :

شباهت ها:

هیچ یک از دو متدولوژی مدیریت ناسازگاری را در زبان مدلسازی ندارند. هر دو متدولوژی مدلسازی را در وجوه مختلف ساختاری، رفتاری و وظیفه ای و همچنین در سطوح انتزاع مختلف انجام می دهند.

تفاوت ها:

متدولوژی UP از پکیج های UML برای قطعه بندی و لایه بندی مدل ها نیز استفاده می کند اما متدولوژی Booch از ساختار هایی برای مدیریت پیچیدگی استفاده نمی کند.

مقایسه:

زبان مدلسازی متدولوژی UP مدیریت پیچیدگی را بهتر از زبان مدلسازی Booch انجام می دهد که این برتری آن محسوب می شود.

مقایسه متدولوژی UP و OMT :

شباهت ها:

هیچ یک از دو متدولوژی مدیریت ناسازگاری را در زبان مدلسازی ندارند. هر دو متدولوژی مدلسازی را در وجوه مختلف ساختاری، رفتاری و وظیفه ای و همچنین در سطوح انتزاع مختلف انجام می دهند.

تفاوت ها:

متدولوژی OMT از سازوکار خاصی برای مدیریت پیچیدگی استفاده نمی کند و مدیریت پیچیدگی آن با وجود مدل ها در وجوه مختلف و سطح انتزاع مختلف در فاز ها صورت می گیرد در حالی که متدولوژی UP از پکیج‌های UML برای قطعه بندی و لایه بندی مدل ها نیز استفاده می کند.

مقایسه:

متدولوژی UP بدلیل استفاده از پکیج های UML برای قطعه بندی و لایه بندی مدل ها، مدیریت پیچیدگی را بهتر از متدولوژی OMT انجام می دهد.