عنوان: **UML و VS2010**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۵/۰۹ ۱۸:۵۴:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: UML

یک دوره‌ی نسبتا مفصل مدلسازی سیستم و سپس ترسیم نمودارهای UML مرتبط با آن به کمک ابزارهای جدید VS2010 اخیرا به سایت channel9 اضافه شده است که لیست آن به شرح زیر است:

UML with VS 2010 Part 1: Brainstorming a Project
UML with VS 2010 Part 2: Organizing Features Into Use Cases
UML with VS 2010 Part 3: Modeling the Business Domain
UML with VS 2010 Part 4: Capturing Business Workflows
UML with VS 2010 Part 5: Architecting an Application
UML with VS 2010 Part 6: Designing a Project's Physical Structure
UML with VS 2010 Part 7: Sketching Interactions with Sequence Diagrams
UML with VS 2010 Part 8: Revealing Responsibilities with Class Diagrams
UML with VS 2010 Part 9: Organizing and Managing Your Models

نظرات خوانندگان

نویسنده: shahin
تاریخ: ۰۸:۳۵:۴۹ ۱۳۸۹/۰۵/۱۰

سلام آقای نصیری خسته نباشید.
ممنون از معرفی لینک های مفید.
هیچ راهی برای دانلودشون نیست؟
ویدیو مشابه که بشه دریافت کرد همم هست؟
با تشکر.

نویسنده: وحید نصیری
تاریخ: ۰۸:۵۷:۳۱ ۱۳۸۹/۰۵/۱۰

چرا. تمامشون قابل دریافت هستند. در همان صفحه به عبارت "Media Downloads" دقت کنید. یک منو باز می‌شود که فرمت‌های مختلفی را جهت دانلود ارائه می‌دهد.

نویسنده: wild_honey
تاریخ: ۲۲:۲۳:۴۵ ۱۳۸۹/۰۵/۱۶

سلام آقای نصیری
این ویدیو ها عالی بودن !
واقعا ممنونم احتمال داره از EA کلا سوئیچ کنم رو 2010 !!

عنوان: آموزش مهندسی نرم افزار و UML - جلسه اول
نویسنده: علی قمشلویی
تاریخ: ۱۳۹۱/۰۴/۰۱ ۲۱:۱۶
آدرس: www.dotnettips.info
برچسب‌ها: UML, مهندسی نرم افزار

آموزش مهندسی نرم افزار و UML
جلسه اول:

اولین قدم در تولید و توسعه نرم افزار داشتن یک نگرش سیستمی به بسته یا محصول نرم افزاری می‌باشد. اما چرا ما باید نرم افزار را به عنوان یک سیستم در نظربگیریم ؟
جواب این سئوال را باید از تعریف تئوری سیستم و خصوصیاتی که یک سیستم دارا می‌باشد استخراج کنیم.

تئوری سیستم‌ها

دانشی برای سهولت کار با سیستم‌ها و بررسی دقیق این مفهوم است ؛ در واقع تئوری سیستم‌ها روشی برای شناخت محیط اطراف یا روشی برای شناخت دنیای واقع می‌باشد .

از تعریف فوق می‌توان نتیجه گرفت :
برنامه نویسان برای ساخت برنامه هایی که با نیاز کاربران همسو باشد ، نیاز به شناخت محیطی دارند که کاربران در آن فعالیت می‌کنند پس برای شناخت محیط باید با دید سیستمی به مسئله نگاه کرد.

خصوصیات مهم سیستم :

1. محیط – Environment: هر سیستم در یک محیط قرار دارد.
2. مرز – Boundary : سیستم‌های موجود در یک محیط توسط مرزها از یکدیگر جدا می‌شوند.
3. ورودی و خروجی – I/O : هر سیستم ورودی هایی را از محیط می‌گیرد و خروجی هایی را به محیط پس می‌دهد.
4. واسط – Interface : امکان محاوره سیستم‌ها در یک محیط را فراهم می‌کند.
5. زیر سیستم – Sub System : هر سیستم می‌تواند حاوی چندین زیرسیستم باشد . زیر سیستم‌ها تمام خصوصیت‌های یک سیستم را دارا می‌باشند.
6. مکانیزم کنترلی – Controller : مهمترین بخش یک سیستم می‌باشد. مکانیزم کنترلی در واقع کنترل کننده تمامی فعالیت‌های انجام شده توسط یک سیستم است . ورودی‌ها از طریق مکانیزم کنترلی دریافت می‌شود و بر اساس آن خروجی هایی به محیط پس داده می‌شود.

نتیجه گیری :

با توجه به خصوصیاتی که در مورد سیستم‌ها مطرح شد به راحتی می‌توانیم دلیل علاقه مندی برنامه -نویسان به نوع نگرش سیستمی را در یابیم ، و جود محیط پیرامون یک سیستم و نحوه تبادل اطلاعات این سیستم با سایر سیستم‌ها در این محیط ، شکستن یک سیستم به چند زیر سیستم برای راحتی مسئله و پیاده سازی آسانتر آن و نیز وجود اینترفیس‌ها برای برقراری محاوره ای استاندارد بین زیر سیستم‌های یک سیستم و همچنین وجود ورودی هاو تصمیم گیری براساس ورودی هاو تولید یک خروجی همه و همه از نکات مورد توجه برنامه نویسان در تولید یک بسته نرم افزاری هستند که هماهنگی کاملی با مفاهیم تئوری سیستم‌ها دارند.

نظرات خوانندگان

نویسنده: هاشم
تاریخ: ۱۳۹۱/۰۴/۰۲ ۲:۶

سلام
دوست عزیز با توجه به قدیمی بودن UML بهتره به سمت مباحثی نظیر اسکرام در Agile ، RUP رفت

نویسنده: علی قمشلویی
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۰:۶

با سلام و تشکر
UML و RUP دو مقوله جدا از یکدیگر می‌باشند در واقع RUP از UML استفاده میکند در ادامه مقالات تمامی این موارد را توضیح خواهم داد و شما هر چه بیشتر با مزایای استفاده از UML آشنا خواهید شد.

نویسنده: صالح
تاریخ: ۱۳۹۱/۰۴/۰۲ ۲۰:۴۴

امید وارم این مطلب رو ادامه بدید.

نویسنده: رضا.ب
تاریخ: ۱۳۹۱/۰۴/۲۱ ۱۰:۲۳

متدولوژی‌هایی که شما میگین(RUP یا متدهای چابک) اصلا ربطی به مفاهیم تحلیلی سیستم که در قالب UML ارائه میشه، نداره. بهتره سنجیده‌تر کامنت گذاشت. UML زبان مشترک مدل کردن سیستمه بین توسعه دهندگان و شاید دیگر ذی‌نفعان سیستم.

نویسنده: اژدری
تاریخ: ۱۳۹۱/۰۶/۱۳ ۱۳:۲۳

دقیقا از نام uml مشخصه که فقط یک زبان مشترکه و ربطی به سایر مسائل نداره ، علت به وجود آمدنش هم ساده است ، برای اینکه هر کسی از ظن خودش یار قضیه نشه و خیال همه راحت باشه که میدونن دارن در مورد چی صحبت می‌کنن و داستان اون فیل در تاریکی هم نشود ، و در ضمن رساندن منظور با عکس خیلی بهتر از یک صفحه نوشته است

عنوان: **آموزش مهندسی نرم افزار و UML - جلسه دوم**
نویسنده: علی قمشلویی
تاریخ: ۱۳۹۱/۰۴/۰۴ ۱۸:۲۲
آدرس: www.dotnettips.info
برچسب‌ها: UML, مهندسی نرم افزار

جلسه دوم :

در جلسه پیش در مورد اینکه چرا یک بسته نرم افزاری را باید به عنوان یک سیستم در نظر بگیریم صحبت کردیم در این جلسه به بررسی سیستم‌های اطلاعاتی می‌پردازیم.
قبل از اینکه به بررسی سیستم های اطلاعاتی بپردازیم به چند مفهوم می‌پردازیم که برای تعریف سیستم‌های اطلاعاتی به آن‌ها نیازمندیم.

· داده – Data : داده خام پردازش نشده ای که از نظر سیستم مفهومی ندارد.
· اطلاعات - Information : داده‌های پردازش شده ای که از نظر سیستم دارای مفهوم خاصی می‌باشند.
· Knowledge : مانند Information دارای مفهوم خاصی هستند اما کمی ساخت یافته‌تر گشته و دارای معنی بیشتری هستند و اغلب در سیستم‌های خبره به کار می‌روند.

تعریف سیستم‌های اطلاعاتی (Information Service)

سیستم هایی هستند که ورودی آنها اطلاعات خام پردازش نشده (Data) و خروجی آنها Information می‌باشد ؛ عمل اصلی این سیستم‌ها پردازش اطلاعات است .

انواع سیستم‌های اطلاعاتی :

1. TPS (Transaction Processing Systems)
عملکرد اصلی TPS ها پردازش اطلاعات است.

2. MIS (Management Information Services)
اطلاعات را برای مدیران سطح بالا پردازش می‌کنند و آ ن‌ها را در تصمیم گیری‌ها یاری می‌دهند.

در ادامه به بررسی مشکلات سیستم‌های اطلاعاتی یا همان بسته‌های نرم افزاری خواهیم پرداخت و راهکاری را که IT برای فایق آمدن به این مشکلات بیان کرده اند را شرح خواهم داد.

برخی مشکلات توسعه سیستم‌های اطلاعاتی (IS) :

1. قیمت پیشنهادی از سوی کارفرما
2. زمان تحویل سیستم غیر معقول باشد
3. هزینه استقرار سیستم بالا باشد
4. تغییر نیازمندی ها
5. کمبود تجربه و تخصص نیروی فنی
6. غیر ممکن بودن پیاده سازی یک سیستم از لحاظ تکنیکی
7. اندازه گیری میزان حرکت پروژه در راستای هدف خود
8. پروژه تا چه اندازه نیازمندی‌های کاربران را پاسخ می‌دهد
9. سختی کار با سیستم
10. سیستم از ورود اطلاعات نامعتبر جلوگیری نکند
11. پیغام‌های خطای نامناسب

12. Help نامناسب
13. غیر قابل اعتماد بودن عملیات‌های سیستم
14. زمان پاسخ گویی نامناسب
15. ...

قبل از اینکه به بیان راهکار IT در این رابطه بپردازیم به تعریفی کوتاه از آن توجه کنید.

IT چیست:
IT راهکاری برای مقابله با مشکلات تولید و توسعه نرم افزار می‌باشد. نیاز به پیاده سازی سیستم‌های اطلاعاتی منجر به پیدایش مفهوم IT شد. پاسخ IT برای فایق آمدن بر مشکلات تولید و توسعه نرم افزار استفاده از متدولوژی است.

در ادامه به بررسی متدولوژی خواهیم پرداخت.

عنوان: **آموزش مهندسی نرم افزار و UML - جلسه سوم**
نویسنده: علی قمشلویی
تاریخ: ۱۳۹۱/۰۴/۰۹ ۱۶:۵۶
آدرس: www.dotnettips.info
برچسب‌ها: UML, مهندسی نرم افزار

جلسه سوم :

در جلسه قبل به بررسی مشکلات تولید و توسعه سیستم‌های اطلاعاتی یا همان بسته‌های نرم افزاری پرداختیم در این جلسه به راهکاری که IT برای فایق آمدن بر این مشکلات پیش روی ما قرار داده یا همان متدولوژی می‌پردازیم.

متدولوژی چیست ؟

متدولوژی در واقع مجموعه ای از روش‌ها ، اصول و قواعدی است که برای قانونمند کردن تولید و توسعه نرم افزار ارائه می‌شود؛می توان گفت متدولوژی فرمولی جهت ساخت نرم افزار می‌باشد یا به عبارت دیگر متدولوژی چرخه حیات نرم افزار را مشخص می‌کند.

- **چرخه حیات تولید وتوسعه نرم افزار یا SDLC(System Development Life Cycle)**

مراحلی را که در طی تولید و توسعه نرم افزار سپری می‌شوند را SDLC می‌گویند.

انواع SDLC

1. چرخه حیات سیستم‌های قدیمی یا TLC

2. چرخه حیات سیستم‌های شی گرا یا OODLC

-(TLC(Traditional Life Cycle

در گذشته به دلیل اینکه اکثر برنامه‌ها بصورت فرآیندگرا یا Process Oriented نوشته می‌شدند از روش TLC استفاده می‌شد . در روش‌های فرآیند گرا تمرکز اصلی بر روی فعالیت‌های سیستم بود در این روش بیشتر از نمودارهای ERD و DFD استفاده می‌شد .

البته اینا اضافه کنم که هنوز هم در بعضی از شرکت‌ها از این روش استفاده می‌شه هر چند که خودشونم نمی‌دونند یکی از دلایل اصلی هم فقر سواد شرکت‌های کار فرما می‌باشد . در ادامه به بررسی یکی از مدل‌های معروف TLC یعنی مدل آبشاری می‌پردازیم.

- مدل آبشاری یا Water Fall :

مدل آبشاری هر چند مدلی قدیمی می‌باشد اما مبنای اساسی مدل‌های شی گرا می‌باشد.

فازهای مختلف مدل آبشاری :

1. مهندسی سیستم یا System Engineering

معرفی نیازمندی‌های کلی و مشخص نمودن کلیات سیستم به صورت سخت افزاری و نرم افزاری و تعاریف اصلی سیستم به طور مثال در پروژه وب سایت از Asp استفاده کنیم یا Php

2. آنالیز نیازمندی‌ها یا Requirement Analysis

در این فاز به نیازمندی‌های کاربران می‌پردازیم یعنی در این فاز ما با چه یا What می‌پردازیم

3. طراحی یا Design

در این فاز ما به دنبال چگونه یا Who می‌رویم یعنی اینکه سیستم چگونه در جهت بر آوردن نیازمندی‌ها گام بردارد.

4. ساخت یا Construction

در این فاز آنچه را که در فاز طراحی مطرح کردیم به کد تبدیل می‌کنیم

5. تست Testing

در این مرحله سیستم از لحاظ کمی و کیفی تست میشوند.

6. نصب یا Installation

7. نگهداری یا Maintenance

این فاز طولانی‌ترین و پرهزینه‌ترین قسمت چرخه عمر یک نرم افزار

معایب مدل آبشاری :

1. مدل آبشاری تکرار بین فازها را در نظر نمی‌گیرد و خروجی هر فاز را قطعی در نظر می‌گیرد که اگر مثلا در فاز طراحی باشیم و یک نیازمندی در نظر گرفته نشده باشد این مدل هیچ راهکاری را برای در ج این نیازمندی در سیستم ارائه نمی‌دهد بعدها برای رفع این مشکل مدل آبشاری با تکرار (Water Fall with Iteration) معرفی گردید.

2. دراین روش هر فاز هنگامی آغاز می‌شود که فاز قبل از خودش به پایان رسیده باشد که این امر مانع از Overlap یا به اشتراک گذاری بین فازها می‌شد.

3. سیستم‌های محاوره ای نیستند و در برابر تغییرات مقاوم نمی‌باشند.

مزایای مدل‌های آبشاری

1. واگذاری هر فاز به یک تیم مشخص

2. پیشرفت پروژه بیشتر به چشم می‌خورد.

در جلسه آینده به بررسی مدل‌های شی گرا خواهیم پرداخت.

نظرات خوانندگان

نویسنده: میثم هوشمند
تاریخ: ۱۳۹۱/۰۴/۱۰ ۰:۱۴

"در این فاز ما به دنبال چگونه یا Who می‌روی "

how

!

نویسنده: رضا.ب
تاریخ: ۱۳۹۱/۰۴/۲۱ ۱۰:۴۷

یه انتقاد دارم که البته نظر شخصیمه. ولی این سه قسمت به‌راحتی میشد تو یه قسمت گفت. و باز هم به‌نظرم بار علمی چنین عنوانی برای یک پست‌های سریالی باید به مراتب غنی‌تر باشد.
با تشکر.