

معماری مقیاس بزرگ

خانم دکتر سحر آدابی

علیرضا سلطانی نشان

۳۰ مهر ۱۴۰۳

فهرست مطالب

۱	پیشگفتار	۱
۲	معرفی	۲
۲	۱.۲ چه زمانی یک پروژه مقیاس بزرگ است؟	۲
۲	۲.۲ یادآوری متدولوژی RUP	۲
۲	۱.۲.۲ منظور از نظم در RUP	۲
۲	۲.۲.۲ چهار فاز اصلی در RUP	۲
۳	۳.۲.۲ منظور از فرسخ شمار چیست؟	۳
۳	۴.۲.۲ محوریت بر روی نیازمندی‌ها	۳
۳	۵.۲.۲ استفاده از برنامه‌نویسی OOP	۳
۴	۳.۲ معماری نرم‌افزار چیست؟	۴
۴	۴.۲ المان‌ها	۴
۴	۵.۲ External Feasible Properties	۴
۴	۶.۲ تفاوت کامپوننت و المان	۴
۴	۷.۲ قابلیت اطمینان یا Reliability	۴
۴	۸.۲ قابلیت استفاده یا Useability	۴
۴	۹.۲ ارائه سریع محصول یا Short time to market	۴
۵	۱۰.۲ تفاوت معماری سازمانی و معماری نرم‌افزار	۵

مجوز

به فایل license همراه این برگه توجه کنید. این برگه تحت مجوز GPLV۳ منتشر شده است که اجازه نشر و استفاده (کد و خروجی/pdf) را رایگان می‌دهد.

۱ پیشگفتار

اگر درس مهندسی نیازمندی‌ها را خوانده باشید، احتمالاً در جریان آن هستید که برای تولید نرم‌افزار بخش‌های زیادی درگیر هستند اما در حالت کلی در درس پیشین دانستیم که در ابتدا بایستی نیازمندی‌های مشتری یا کارفرما را از محصول نرم‌افزار بدانیم، آن را بررسی و

تحلیل کنیم، سند نیازمندی آن را آماده‌سازی کنیم و سپس به دنبال طراحی معماری آن برویم. در این درس به طراحی و پیاده‌سازی سند معماری مقیاس بزرگ یک محصول نرم‌افزاری می‌پردازیم تا فرایند تولید نرم‌افزار را به طور کامل طی کرده باشیم.

۲ معرفی

۱.۲ چه زمانی یک پروژه مقیاس بزرگ است؟

برای اینکه بتوانیم بگوییم که چه پروژه‌ای مقیاس بزرگ محسوب می‌شود، براساس دو استاندارد ایرانی و بین‌المللی می‌توان دو استاندارد را در اینجا مطرح کرد:

- استاندارد مقیاس بزرگ بودن پروژه از نظر دکتر شمس، آن است که پروژه بیشتر از ۶ ماه زمان پیاده‌سازی نیاز داشته باشد و تعداد درخواست‌های ارسالی به آن ۱۲ نفر به بالا باشد.
- استاندارد بین‌المللی مقیاس بزرگ بودن پروژه را زمان یک سال به بالا جهت پیاده‌سازی و تعداد درخواست‌ها را بین ۲۰ تا ۲۲ نفر تعیین می‌کند.

ابتدایی ترین فاز معماری یک محصول نرم‌افزاری مقیاس بزرگ، طراحی و بررسی و آنالیز سناریوهای آن است. سند معماری نرم‌افزار به مجموعه‌ای از سناریوهایی گفته می‌شود که در ازای هر کدام یک راه‌حل مناسب مطرح می‌شود. یکی از نیازمندی‌های بررسی معماری نرم‌افزار مقیاس بزرگ استفاده از متدولوژی RUP^۱ می‌باشد. دلیل اصلی آن این است که می‌توان تمام فرایندهای آن را به همراه Artifactها شخصی‌سازی کرد. در معماری نرم‌افزار می‌توانیم مشخص کنیم که چه اجزایی داریم و این اجزا چگونه با یکدیگر در ارتباط هستند و شامل چه قیدهایی می‌شود. در حقیقت در سند معماری نرم‌افزار نمود خارجی المان‌ها را مطرح می‌کنیم. نحوه در کنار هم چیدن سرویس‌ها را مطرح می‌کنیم اما هیچ وقت در مورد جزئیات اینکه برای مثال از چه الگوریتم‌هایی استفاده می‌کنیم، صحبت نمی‌شود. در این سند علاوه بر نیازهای جاری، در مورد نیازهای آتی نیز صحبت می‌شود که در آینده چقدر باید نرم‌افزار قابلیت گسترش Expandability داشته باشد.

۲.۲ یادآوری متدولوژی RUP

این متدولوژی به عنوان یک متدولوژی توسعه نرم‌افزار اجایل، به دلیل قابل تکرار بودنش در نظر گرفته شده است. این روش مهندسی نرم‌افزار از یک سیستم انعطاف پذیر و سازگار در فرایند توسعه نرم‌افزار استفاده می‌کند که در برگیرنده انجام تنظیمات و تکرار دوره‌های مهندسی نرم‌افزار است تا زمانی که محصول به نیازمندی‌های مطرح شده و اهداف برسد [۱].

۱.۲.۲ منظور از نظم در RUP

منظور از نظم در حقیقت نمودهایی می‌باشد که در فرایند توسعه نرم‌افزار مورد استفاده قرار می‌گیرد، در حقیقت نظم، مدل‌سازی حرفه‌ای را نشان می‌دهد. این نظم‌ها به ما کمک می‌کنند که چه زمانی چه Activityهایی را باید به چه میزان در چه بازه‌هایی انجام دهیم و خروجی مورد نظر ما چیست؟

برای مثال در فرایند تحلیل نیازمندی پروژه، نظم نیازمندی، خروجی فازهای آن است که به شکل مدل‌های Use case diagram و سند معماری نرم‌افزار کشیده و نوشته شده است.

۲.۲.۲ چهار فاز اصلی در RUP

۱. فاز آغاز یا Inception: در این فاز تمام نیازمندی‌ها جمع‌آوری می‌شود و مقیاس پروژه در آن بدست می‌آید.

۲. فاز توسعه یا Elaboration: طراحی سیستم و تحلیل دقیق‌تر نیازمندی‌ها صورت می‌گیرد.

^۱Rational Unified Process

(آ) استفاده از مدل‌سازی‌ها و کشیدن دیاگرام‌ها

(ب) کشیدن مدل usecase: کاربرد بزرگی برای مشتری (کارفرما) و طراح دارد و برای هر دو طرف قابل فهم می‌باشد. در این نوع نمودار افعال و نیازمندی‌های functional مطرح می‌شود. انتظارات در مورد سیستم در اینجا مورد بحث قرار می‌گیرند.

- اینکه کاربرد بتواند زیر ۲ ثانیه احراز هویت شود مربوط به نیازمندی‌های non-functional می‌باشد.

- شامل دو سند می‌شود:

- سند Usecase که انتظارات سیستم را مشخص می‌کند.

- سند معماری که function و non-functional را در بر می‌گیرد.

(ج) طراحی Class diagram

(د) طراحی Sequence diagram

۳. فاز ساخت یا Construction: در این فاز کد نویسی و ارزیابی کدهای نوشته شده صورت می‌گیرد.

۴. فاز استقرار یا Deployment: در این فاز نرم‌افزار آماده شده است و در بستری مناسب به کاربران نهایی^۲ ارائه می‌شود که نیازمند آموزش‌های لازم می‌باشد.

محبوبیت استفاده از متدولوژی RUP به خاطر آن است که کاملاً به صورت جامع سیستم را در بر می‌گیرد.

۳.۲.۲ منظور از فرسخ‌شمار چیست؟

فرسخ‌شمار یا Milestone در هر کدام از فازها مشخص می‌شود که در حقیقت در مورد تعیین یک بازه زمانی مشخص صحبت می‌کند. در آن می‌توانیم ببینیم که در فازهای قبلی چه کارهایی بایستی انجام می‌شده، آیا آن‌ها را انجام داده‌ایم و اگر انجام نداده‌ایم یا مشکلی در آن وجود دارد آن فاز را تکرار می‌کنیم تا به انتهای آن برسیم که به نحوی تسک یا وظیفه را ببندیم.

نکات

- ساده‌ترین سند در میان این ۴ فاز، سند استقرار می‌باشد.

- معماری مقیاس‌پذیر (بزرگ) یک پروژه نرم‌افزار دو بُعد پویا و ثابت دارد.

- در مورد ارزیابی کارایی و آزمون نرم‌افزار گفتنی است که هر توسعه‌دهنده مسئول Quality control بخش خودش است.

- تکرارها n تا هستند مدیر پروژه یا طراح سیستم باید به ما تعداد تکرارها را به صورت تقریبی بگوید.

۴.۲.۲ محوریت بر روی نیازمندی‌ها

متدولوژی RUP تأکید زیادی روی شناسایی و مدیریت نیازمندی‌ها را دارد و به تیم‌ها کمک می‌کند تا نیازمندی‌های کلیدی پروژه را به خوبی درک و پیاده‌سازی کنند.

۵.۲.۲ استفاده از برنامه‌نویسی OOP

این متدولوژی به طور گسترده از ۴ اصل شیء‌گرایی استفاده می‌کند و به توسعه‌دهندگان اجازه می‌دهد که کدهای قابل استفاده مجدد و مدیریت فاکتورهای انعطاف‌پذیری را ایجاد کنند.

^۲End users

۳.۲ معماری نرم افزار چیست؟

معماری نرم افزار یک تعریف واحد ندارد. معماری نرم افزار یک برنامه یا یک سیستم محاسباتی می باشد. یک ساختار یا مجموعه ساختارهایی است از سیستم مورد نظر ما که متشکل از المان های کامپیوتری است و نمود خارجی یک چیز (المان) می باشد و ارتباطات بین آن ها را در بر می گیرد. هیچگاه نمی توان نرم افزاری نوشت که معماری نرم افزار نداشته باشد. برای مثال از معماری MVC در نرم افزار خود استفاده کرده ایم. نرم افزاری وجود ندارد که معماری نداشته باشد. اگر بگوییم نرم افزاری معماری ندارد در حقیقت علم معماری به کار گرفته شده را نمی دانیم که آن را بی معماری می نامیم. برای مثال معماری کلاینت سرور که براساس نیازمندی های نرم افزاری بیان می شود که چه بخش هایی سمت سرور باشد چه بخش هایی سمت کلاینت.

۴.۲ المان ها

بخش های یک سیستم نرم افزاری را گویند برای مثال یک نرم افزار واحد مانیتورینگ، واحد زمان بندی، واحد بررسی درخواست ها و غیره را دارد.

۵.۲ External Feasible Properties

آن بخش چه وظیفه ای را باید انجام دهد و آن بخش آن وظیفه را در حال انجام است یا خیر؟ جزئیات مربوط به المان های درگیر در بخش معماری در External Feasible Properties مطرح نمی شود.

۶.۲ تفاوت کامپوننت و المان

وقتی در مورد کامپوننت می گوئیم در حقیقت چیزی است که می خواهیم آن را پیاده سازی کنیم. المان کامپوننتی است که قسمت اجرایی را برای آن در نظر نگرفته ایم.

۷.۲ قابلیت اطمینان یا Reliability

یک سیستمی که در زمان مشخص درست کار کند به شرطی که در زمان $T - x$ درست کار کرده باشد.

۸.۲ قابلیت استفاده یا Useability

سیستمی که کار آمد باشد برای آن دسته از افرادی که سیستم را حاضر و آماده کرده ایم. به گونه ای که با ظاهر مناسب کار کردن با آن نیز آسان باشد.

۹.۲ ارائه سریع محصول یا Short time to market

قابلیت یا Feature مجموعه ای از توابع نرم افزاری است که وقتی در بازار ارائه می شود، واقعاً کار می کند.

نکات

- تا آنجایی که می شود هزینه ها را باید کاهش بدهیم و بیشترین هزینه ها را ما در بخش توسعه نرم افزار خواهیم داشت. همیشه باید کارمندان را مشغول توسعه نگهداریم تا باعث از دست رفتن هزینه ها نشود.
- مشکل معمار آن است که حجم زیادی از نیازمندی های نرم افزاری را در حال بررسی است که نسبت به هم در تضاد هستند.
- بخشی از وظایف اصلی معماری نرم افزار مشخص کردن استک های نرم افزاری می باشد. بخش دیگری از آن این است که بررسی کند این موارد توسط تیم اجرا و استفاده می شود یا خیر (Follow up)

۱۰.۲ تفاوت معماری سازمانی و معماری نرم افزار

معماری سازمانی و معماری نرم افزار با یکدیگر متفاوت است. در معماری سازمانی، چارت سازمانی آن مشخص کننده محدوده و کلیت هر قسمت آن سازمان می باشد.

- [1] Study.com. Rational unified process | rup definition, methodology & examples. [https://study.com/academy/lesson/what-is-the-rational-unified-process-methodology-tools-examples.html#:~:text=Rational%20Unified%20Process%20\(RUP\)%20is,requirements%20and%20objectives%20are%20met.,](https://study.com/academy/lesson/what-is-the-rational-unified-process-methodology-tools-examples.html#:~:text=Rational%20Unified%20Process%20(RUP)%20is,requirements%20and%20objectives%20are%20met.,) 11/21/2023.