

تکرار و توسعه تدریجی:

فرایند تولیدسنتی شامل مراحل تحلیل نیازمندیها، تعیین مشخصات، طراحی، پیاده سازی، آزمایش، نگهداری و تکامل است. ویژگی اصلی این روش ترتیبی بودن آن است. در این روش با گذشت زمان، هزینه ریسک بالاتر می رود. این روش برای پروژه های کوچک یا کوتاه مدت مناسب است و برای پروژه های بزرگ، آنها را به چند زیرپروژه کوچک تر می شکنیم و سپس در هر کدام مدیریت ریسک را قرار می دهیم.

ویژگی های روش تکرار و توسعه تدریجی:

- (1) تشخیص زودهنگام خطاها
- (2) تشخیص زودهنگام ناسازگاریها
- (3) نظارت کاربربر روند پیشرفت پروژه
- (4) تمرکز تیم توسعه روی قسمت های مهم تر پروژه
- (5) امکان تشخیص روند پیشرفت پروژه با آزمایش تکراری و مستمر
- (6) توزیع متوازن بار کاری تیم روی چرخه تولید پروژه

مدیریت نیازمندیها:

نیازمندی عبارتست از شرطی یا قابلیت که سیستم باید دارای آن باشد. در طول توسعه، نیازمندیها در حال تغییرند. پس وظایف مدیریت تغییرات به صورت زیر است:

- (1) دریافت، سازماندهی و مستندسازی عملکرد سیستم
- (2) اعمال تغییرات مطلوب روی نیازمندیها
- (3) ردیابی و مستندسازی

استفاده از معماری مبتنی بر مولفه ها:

معماری سیستم عبارتست از تعیین ساختار کلی سیستم و روش هایی که این ساختار را قادر به تامین کلیه ویژگی های کلیدی سیستم می سازد.

معماری سیستم شامل تصمیم گیری در:

- (1) نحوه سازماندهی سیستم نرم افزاری
- (2) انتخاب عناصر ساختاری و مشخص کردن رفتار آنها
- (3) سازماندهی این عناصر در گروه های بزرگتر
- (4) سبک معماری مورد استفاده

ویژگی استفاده از معماری مبتنی بر مولفه ها:

- (1) داشتن یک معماری انعطاف پذیر
- (2) افزایش قابلیت استفاده مجدد
- (3) مولفه ها یک پایه خوب و مناسب برای مدیریت پیکربندی

مدلسازی تصویری نرم افزار:

مدل عبارتست از یک توصیف ساده شده با توجه به یک نگرش معین از سیستم.

مدلسازی دارای ویژگی های زیر است:

- (1) امکان توصیف سیستم
- (2) طراحی روشن و صریح از سیستم
- (3) تشخیص معماریهای غیرقابل انعطاف و واحدبندی نشده
- (4) نرم افزار خوب حاصل مدل‌های با کیفیت بالاست.

بررسی کیفیت نرم افزار:

برای رسیدن به یک نرم افزار با کیفیت عالی باید فرایند تشخیص کیفیت به صورت پیوسته از همان مراحل اولیه توسعه نرم افزار به اجرا درآید.

هزینه اصلاح خطاها به صورت نمایی رشد می کند، پس باید آنها را زودتر تشخیص دهیم.

ویژگی های بررسی کیفیت نرم افزار:

- (1) فرایند تشخیص پیشرفت پروژه مبتنی بر واقعیت ها
- (2) آشکار کردن ناسازگاری موجود بین نیازمندیها
- (3) کشف زودهنگام خطاها
- (4) کاهش هزینه اصلاح خطاها

مدیریت پیکربندی:

مدیریت پیکربندی، هنر تشخیص، سازماندهی و کنترل تغییراتی است که برای نرم افزار در مدت کارکرد خود رخ می دهد.

ویژگی های مدیریت پیکربندی:

- (1) روش سیستماتیک و قابل تکرار برای کنترل تغییرات نرم افزار
- (2) مدیریت کنترل انتشار تغییرات
- (3) کاهش تداخل بین کار تیم توسعه دهنده
- (4) استفاده از نرخ تغییر برای تشخیص وضعیت فعلی پروژه

متدولوژی های توسعه نرم افزار:

(1) متدولوژی های سنگین وزن: این نوع معمولاً مستندات، محصولات و فرآورده های بیشتری تولید می نمایند.

(2) متدولوژی های سبک وزن: این نوع تاکید بر تولید محصول نهایی می کنند نه فرآورده ها و مستندات بسیار

تفاوت متدولوژیهای سنگین وزن و سبک وزن:

(1) روش اجرا

- روش های سبک وزن به صورت سازگار عمل می کنند.
- روش های سنگین وزن به صورت پیشگو عمل می کنند.

(2) معیار موفقیت

- در روش های سبک وزن دستیابی به ارزش کاری
- در روش های سنگین وزن پیش رفتن در راستای طرح اولیه

(3) اندازه پروژه

- در روش های سبک وزن کوچک
- در روش های سنگین وزن بزرگ

(4) سبک مدیریت

- در روش های سبک وزن غیر متمرکز و آزاد
- در روش های سنگین وزن مطلق و استبدادی

تفاوت متدولوژیهای سنگین وزن و سبک وزن:

(5) مستند سازی

- در روش های سبک وزن بسیار محدود
- در روش های سنگین وزن کامل و جامع

(6) تعداد چرخه ها

- در روش های سبک وزن بسیار زیاد اما زمان آنها کوتاه
- در روش های سنگین وزن کم ولی زمان آنها زیاد

(7) اندازه تیم

- در روش های سبک وزن کوچک
- در روش های سنگین وزن بزرگ

(8) برگشت سرمایه

- در روش های سبک وزن خیلی زود در طول پروژه برمی گردد.
- در روش های سنگین وزن باید تا انتهای پروژه صبر کرد.

:Unified Software Development Process

USDP فرآیند تولید مهندسی نرم افزار است که یک روش سیستماتیک و منظم برای ترتیب انجام فعالیت ها در یک پروژه نرم افزاری را پیشنهاد می کند.

هدف این فرآیند، توسعه نرم افزارهایی با کیفیت عالی که علاوه بر برآوردن نیازهای کاربران خود، در زمان و با هزینه پیش بینی شده تولید شوند.

:Unified Software Development Process

USDP از مدل شی گزایی حمایت می کند و از روش های مدرن توسعه نرم افزار مانند:

- استفاده از روش تکرار و توسعه تدریجی
- معماری مبتنی بر مولفه ها
- راهبری بر موارد کاربری
- مدلسازی نرم افزار
- کنترل تغییرات
- بررسی کیفیت نرم افزار
- پشتیبانی می کند.

محورهای USDP:

- راهبری بر مبنای موارد کاربری:

مورد کاربری عبارتست از دنباله ای از عملیات که یک سیستم انجام می دهد تا یک نتیجه قابل مشاهده و ارزشمند برای کاربر فراهم کند.

- محوریت قرار دادن معماری:

یکی از بهترین تجربیات توسعه نرم افزار است.

- استفاده از روش تکرار و توسعه تدریجی:

سبب دو بعدی بودن این فرآیند توسعه شده است.

توسعه USDP:

USDP به عنوان ابزاری برای توسعه جدید همچون RUP مورد استفاده قرار گرفته است. در واقع شرکت Rational با سفارشی نمودن فرآیند USDP و افزودن ابزارها، راهنماها و ... محصولی به نام RUP را تولید نموده است.