

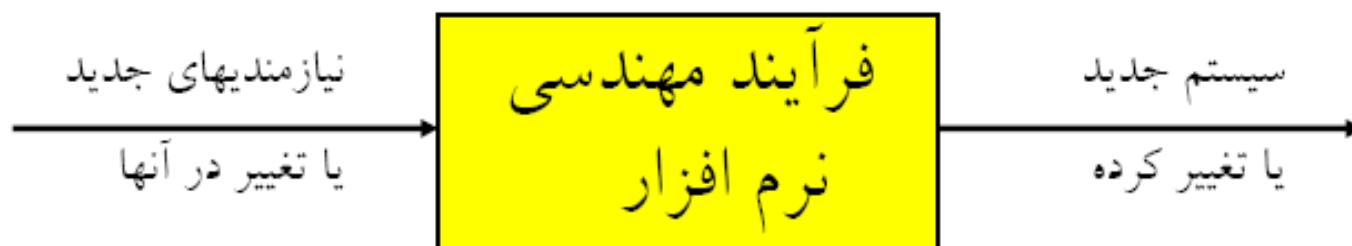
Object Oriented Design Course

RUP (RATIONAL UNIFIED PROCESS)

<http://www.students.madadyar.com>

✂ فرآیند تعریف میکند که چه کسی، چه کاری را در چه هنگام، چگونه به منظور رسیدن به یک هدف انجام دهد.

✂ در مهندسی نرم افزار هدف ساختن یک محصول نرم افزاری یا بهبود یک نمونه موجود است.



✘ RUP فرآیند تولید و توسعه نرم افزار می باشد:

✘ فعالیت ها و مسئولیت ها را به ترتیب جهت توسعه نرم افزار مشخص می کند.

✘ هدف آن حصول اطمینان از تولید نرم افزار با کیفیت بالا براساس نیازهای کاربران و با توجه به زمان و بودجه در دسترس می باشد.

✘ یک راهکار قانونمند برای انتساب وظایف و مسئولیت ها

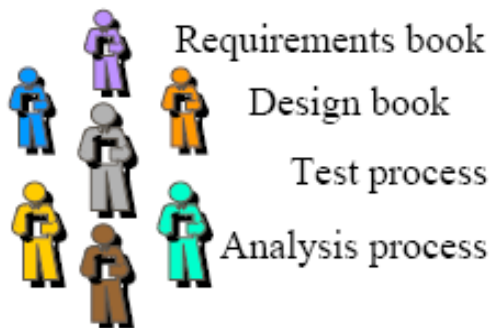
✖ تضمین کیفیت نرم افزار

✖ برآورده شدن نیاز های کاربر

✖ قابل تخمین بودن هزینه و زمان تولید

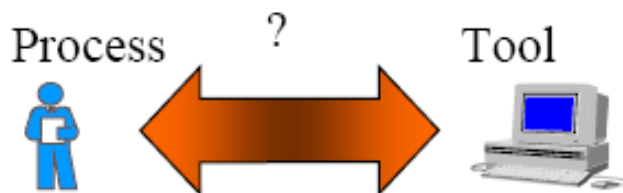


❧ عدم پیروی از فرایند مشخص، یا پیروی از
فرآیندهای خودساخته



❧ استفاده از فرآیندها و زبانهای
متفاوت

❧ عدم پشتیبانی خوب (یا درست) توسط
ابزار



- ✗ یکپارچه سازی فرآیندهای مختلف در قالب یک فرآیند سازگار.
- ✗ تولید و نگه داری مدل بجای مستندات کاغذی.
- ✗ استفاده کارا و مفید از UML.
- ✗ قابلیت استفاده در بازه وسیعی از برنامه ها.



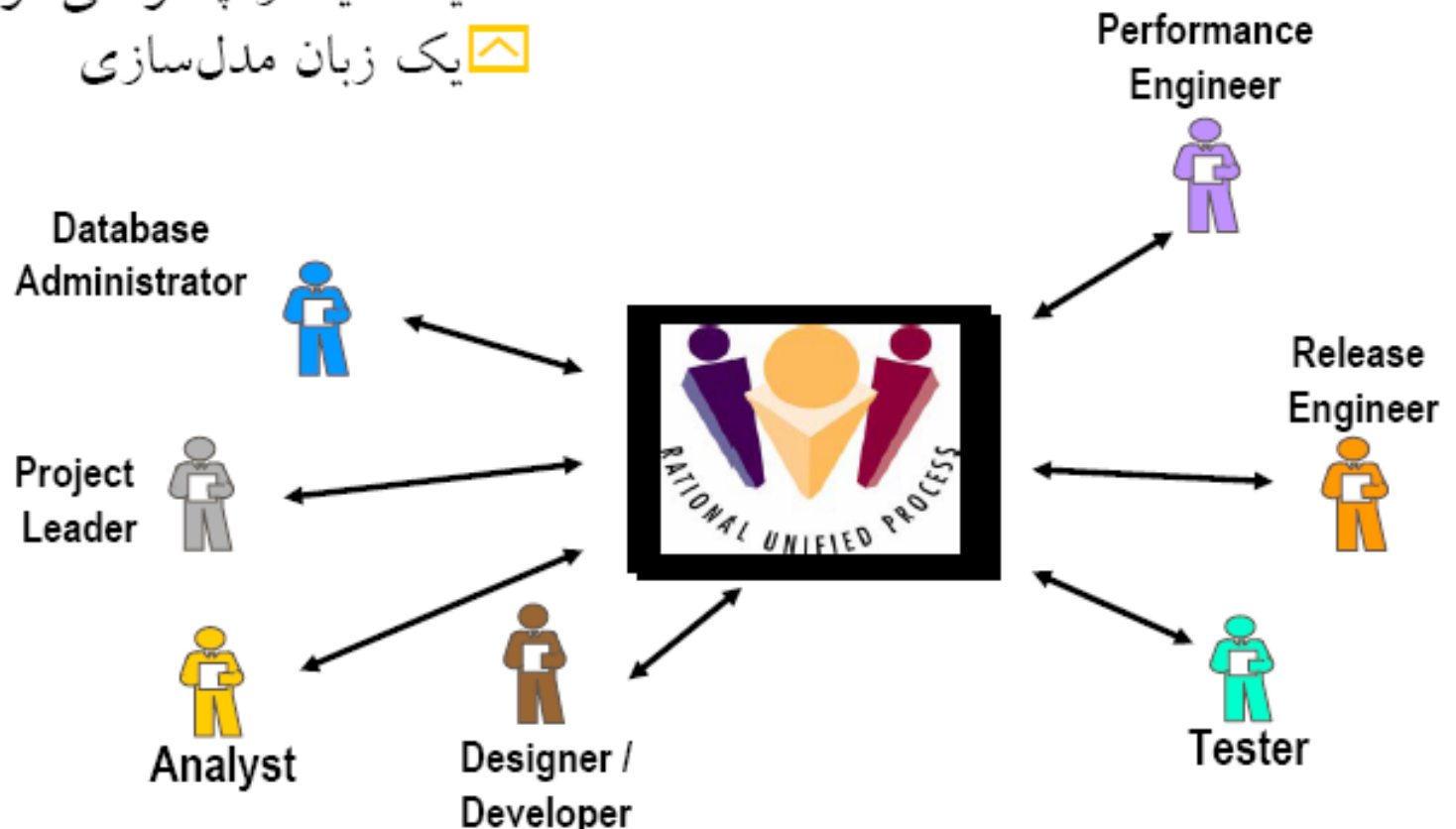
✂ استفاده مشترک اعضا از :

یک پایگاه دانش

یک فرایند

یک دید از چگونگی تولید نرم افزار

یک زبان مدل سازی



✖ ۱. بی درنگ و پیوسته به ریسک ها حمله کنید!

+ به جای اینکه صبر کنید تا ریسک های تجاری ، فنی و ... در مراحل بعدی مشخص شوند آنها را هر چه زودتر شناسایی کرده و اقدامات لازم را انجام دهید.

✖ مثال از ریسک:

- + بر اساس تجارب قبلی نگران هستید فلان سازمان درک درستی از نیازمندیها ندارد و ممکن است بعد از بتا درخواست تغییر کند.
- + نمی دانید چگونه می توان سیستم قدیم را با سیستم جدید یکپارچه نمود.
- + در تولید تحت Net. مشکل دارید و

✖ ۲. اطمینان حاصل کنید که محصول با ارزشی به مشتری تحویل می دهید.

+ نیازمندی ها طوری مستند شوند تا برای مشتری قابل فهم باشند.

+ در مراحل طراحی و پیاده سازی و آزمون نیز روی نیازمندیها کار کنید تا مطمئن شوید محصول مورد نظر مشتری ایجاد شود.

✖ ۳. روی نرم افزار قابل اجرا تاکید کنید.

+ مستندات، طرح ها و ... همگی مفید هستند ولی پیشرفت واقعی را نشان نمی دهند. چرا که هدف اصلی پروژه نیستند و در مقایسه با « کد » در درجه دوم قرار دارند .

+ کد قابل اجرا (آزمایش شده) بهترین شاخص پیشرفت است.

✖ ۴. تغییرات را هر چه زودتر اساس کار خود کنید.

+ بسیاری از ریسک های یک پروژه در اثر طراحی و پیاده سازی زود هنگام در پروژه کاهش می یابد.

+ ایجاد هر چه سریعتر یک معماری پایدار ارتباطات را تسهیل و تغییرات آتی را کاهش می دهد.

✖ ۵. سیستم را با استفاده از کامپوننت ها بسازید.

+ سیستمی که با کامپوننت ها ساخته می شود در مقابل تغییرات بسیار انعطاف پذیر بوده و هزینه نگه داری آن بسیار پایین تر است.

+ به کارگیری کامپوننت ها در ساخت سیستم استفاده مجدد از دارایی ها موجود و در دسترس ساده تر می باشد و برنامه های با کیفیتی ساخته می شود.

✖ ۶. کار را به صورت تیمی و گروهی انجام دهید.

+ تولید و توسعه نرم افزار یک کار گروهی است.

+ تولید به روش تکرار بر ارتباطات گروهی خوب و روحیه تیمی استوار است که در آن هر عضو تیم خود را مسئول و پاسخگو بداند.

× ۷. کیفیت را یک اصل در نظر بگیرید نه یک فرع!

+ تضمین کیفیت تنها بر عهده تیم آزمون نیست و همه اعضای تیم و تمام قسمت های چرخه حیات عهده دار آن هستند.

+ تولید به روش تکرار بر اساس آزمونهای زودهنگام و خودکار کردن این آزمونها تاکید دارد و به همین جهت استفاده از این روش کیفیت محصول نهایی را بالا می برد.

رویه های مناسب در RUP

- ✗ تولید تکراری و تکاملی
- ✗ مدیریت نیازمندی ها
- ✗ استفاده از معماری مبتنی بر اجزاء
- ✗ مدل سازی بصری نرم افزار
- ✗ تضمین کیفیت نرم افزار
- ✗ کنترل تعییرات نرم افزار



✖ ۸. یکی از خصوصیات مهم RUP توالی و تکرار می باشد.

+ در رو شهای سنتی تحلیل و طراحی یک سیستم نرم افزاری، مراحل کار (work flow) به صورت ترتیبی بود.

+ یعنی انجام هر کار یک و فقط یک بار انجام می شد. (waterfall)

✖ در این روش کلیه مسایلی که هنگام تجزیه تحلیل سیستم از قلم افتاده است در هنگام پیاده سازی سیستم رو شده و باعث یک توقف در روند پیشرفت پروژه می گردند و در ادامه کار اختلال به وجود می آورد.

+ راه حل داشتن feedback به مراحل و کامل شدن تدریجی.

✖ در پایان هر تکرار یک release از نرم افزار تولید می شود.

مزایای روش تولید با تکرار

تولید به روش تکرار با نیازمندیهای متغیر سازگار است.

در تولید به روش تکرار یکپارچه سازی به منزله یک اتفاق بزرگ در آخر پروژه نیست.

در این روش ریسک ها در همان ابتدا کشف می شوند.

در تولید می توان تغییرات تاکتیکی داد. (مثلا قابل رقابت کردن محصول)

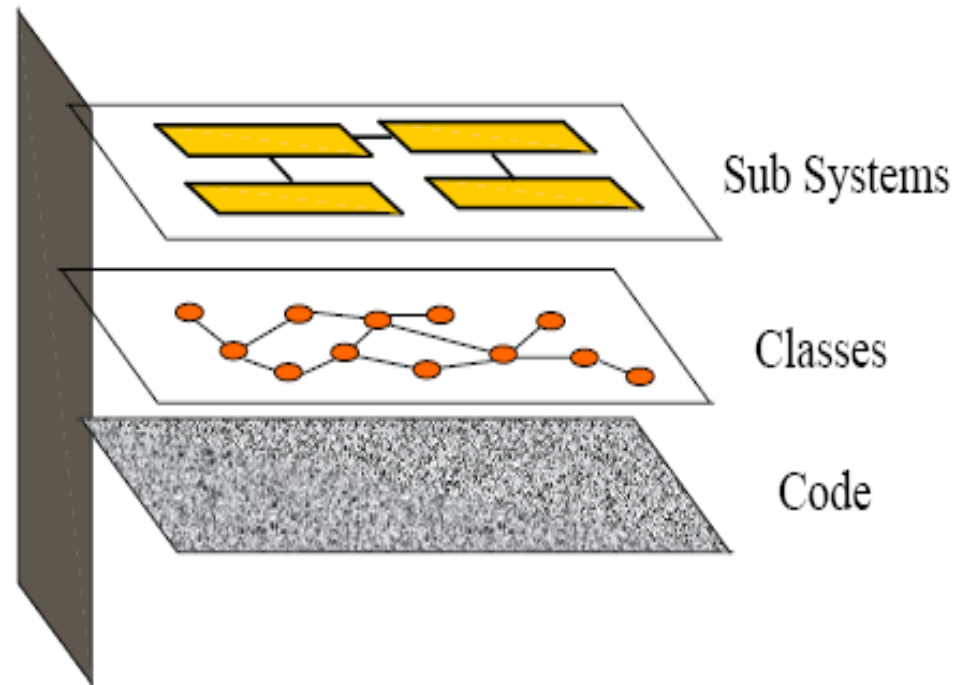
استفاده مجدد آسان می شود.

نقص ها در چندین تکرار کشف و رفع می شوند.

حفظ سازگاری بین طراحی و پیاده سازی



مدل سازی بصری
سطوح تجرد را
افزایش میدهد



تضمین کیفیت نرم افزار

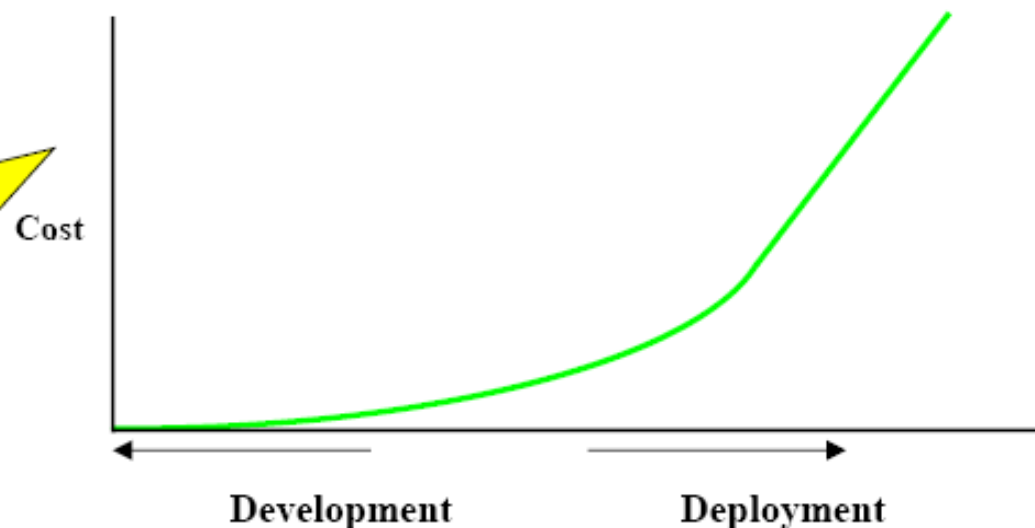
❧ تولید آزمون‌هایی برای هر سناریوی کلیدی تا اطمینان حاصل شود که تمام نیازمندی‌ها بدرستی پیاده‌سازی شده است.

❧ ارزیابی کارایی و قابلیت اطمینان

❧ تضمین قابلیت اطمینان - نشت حافظه و گلوگاه‌ها

❧ تست و آزمایش هر تکرار - تست خودکار

هزینه یافتن و رفع
مشکلات نرم افزار
پس از تحویل ۱۰۰ تا
۱۰۰۰ برابر میشود



الگوی کاری



Disciplines

Business Modeling

Requirements

Analysis & Design

Implementation

Test

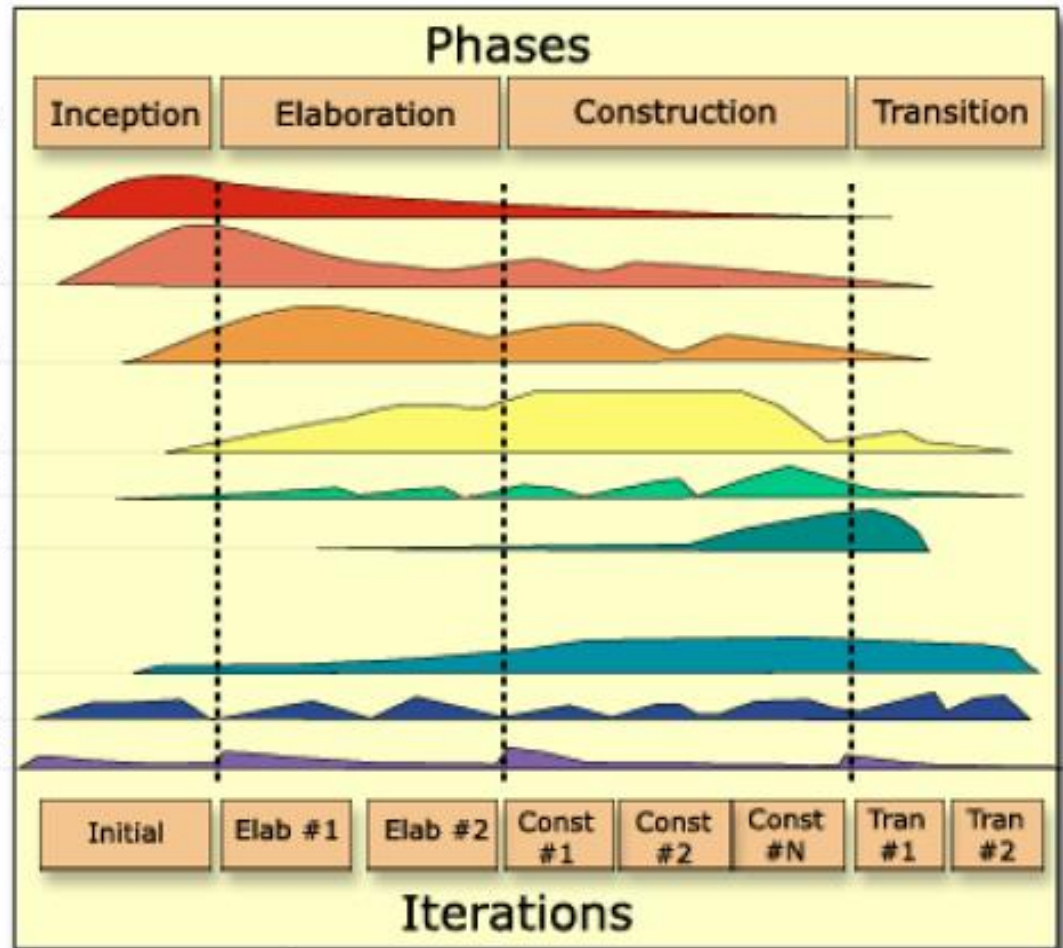
Deployment

Configuration

& Change Mgmt

Project Management

Environment



زمان



✘ محور افقی یا بعد زمان که فازهای چرخه عمر فرآیند را به ترتیب در طول زمان نشان می دهد.

✘ محور عمودی که دیسپلین ها یا الگوهای کاری را نشان می دهد.

+ **بعد اول** جنبه دینامیکی فرآیند را نمایش می دهد، که در قالب فازها (phases) تکرار و توالی (iteration) و نقاط کلیدی (milestone) را تشریح می کند.

+ **بعد دوم** جنبه استاتیکی فرآیند را نمایش می دهد که در قالب اجزای فرایند، الگوهای کاری، فعالیتها، جریانهای کاری محصولات و نقشها (roles) تشریح می شود.

عناصر مدل سازی در RUP

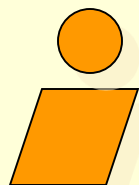
نقش ها - چه کسی کار را انجام دهد؟ (Who)

فعالیت ها - چگونه کار انجام می شود؟ (How)

فرآورده ها - حاصل کار باید چه باشد؟ (What)

نمودارهای گردش کار - کار در چه زمانی باید انجام شود؟ (When)

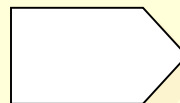
نمونه ای از تعیین نقش ها و فعالیت ها



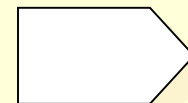
تحلیلگر فرآیند
کسب و کار



ارزیابی سازمان
مورد نظر



ایجاد فهرست
لغات رایج



تنظیم اهداف
سازمانی

فازهای RUP

Inception ✕

+ تعریف حوزه مساله

Elaboration ✕

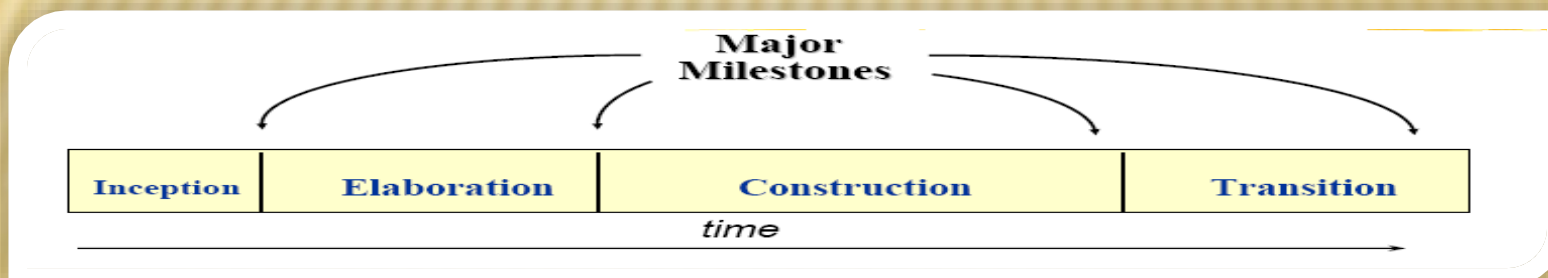
+ مشخص کردن ویژگی ها و پایه گذاری معماری (فاز جزییات)

Construction ✕

+ ساختن محصول

Transition ✕

+ انتقال محصول به محیط کاربر



در پایان هر فاز یک ارزیابی انجام
می گیرد، تا مشخص شود که آیا
اهداف فاز بدست آمده اند یا نه .

اگر نتیجه ارزیابی رضایت بخش
بود، پروژه به فاز بعدی حرکت
می کند.

فاز شناخت (INCEPTION)

✗ هدف اصلی این فاز انجام یک امکان سنجی و شناخت کامل از وضعیت سیستم موجود ، در کنار کسب رضایت کلیه ذینفعان در مسیر انجام پروژه می باشد.

✗ ایجاد محدوده و شرایط مرزی نرم افزار:

+ شامل چشم انداز عملیاتی ، معیارهای پذیرش و تعیین اینکه چه چیزی در محصول باشد و چه چیزی نباشد.

✗ تخمین هزینه کلی و زمانبندی کلی پروژه

✗ برآورد ریسک های موجود و بالقوه

✗ تدارك محیط پشتیبانی برای پروژه

خروجی ها (ARTIFACT)

- × چشم انداز (vision)
- × مورد کسب و کار (Business Case)
- × لیست ریسک ها (Risk Lists)
- × برنامه توسعه نرم افزار (Software Development Plan)
- × برنامه تکرار (Iteration Plan)
- × ابزار (Tools)
- × لغت نامه (Glossary)
- × مدل قالب های کاربردی (Use Case Model)
- × هر کدام از این خروجی ها ، با انجام یکی از الگوهای کاری یا دیسیپلین های RUP ایجاد می شود.

جزئیات ELABORATION

- ✘ هدف این فاز ایجاد معماری سیستم است تا پایه مستحکمی را برای فعالیت های طراحی و اجرا در فاز ساخت فراهم کنیم .
- ✘ معماری مهم ترین نیازمندیها یعنی آنهاییکه بیشترین تاثیر را بر معماری سیستم دارند و همچنین ارزیابی ریسک را در بر می گیرد.
- ✘ در این فاز تصمیم گرفته می شود که پروژه مورد نظر قابلیت ورود به فاز ساخت را داراست یا خیر.

اهداف فاز جزییات

✖ تعریف و تأیید اعتبار و معماری سیستم تا حدی که عملی باشد.

✖ بنیانگذاری چشم انداز سیستم.

✖ بنیانگذاری یک طرح صحیح و قابل اجرا برای فاز ساخت .

✖ اثبات اینکه طرح معماری سیستم کلیه نیازها و انتظارات را با هزینه مناسب و در زمان قابل قبول برآورده خواهد کرد.

ARTIFACT

✕ شکل اولیه (proto Type)

✕ مدل طراحی (Design Model)

✕ مدل داده (Data Model)

ساخت CONSTRUCTION

- ✘ در حین انجام این فاز مکرراً محصولات کاملی تولید می شود که آمادگی انتقال برای استفاده کاربران را دارا می باشند.
- ✘ در این فاز طراحی سیستم استخراج شده و فعالیتهای اجرا و تست نرم افزار تکمیل می گردد.
- + در پایان این فاز بررسی می شود که آیا سازمان و کاربران سیستم آمادگی لازم برای عملیاتی شدن سیستم را دارند یا خیر.
- ✘ در خلال روند این فاز تمامی **component** ها پیاده سازی می شوند.

✖ در واقع مدیریت از ساخت یک سری ویژگیهای ذهنی در مرحله inception و elaboration به ساخت یک محصول قابل گسترش در فاز construction و transition می رسد.

✘ بعد از کدنویسی، برنامه نویسان می توانند بر روی جنبه های ویژه تجاری متمرکز شوند و فعالیت های زیر را انجام دهند:

✘ بازیابی کد براساس استانداردهای طراحی و توافقات حاصله در مراحل قبلی.

✘ بازیابی آبجکت ها به منظور تضمین کیفیت.

✘ این فاز زمانی به پایان می رسد که نرم افزار تکمیل شده و تست گردیده است.

انتقال TRANSITION

- ✘ در این فاز نرم افزار در دست کاربران آن قرار می گیرد.
- ✘ ممکن است نیازهای جدیدی برای متعادل کردن سیستم بوجود آید و یا مسایل حل نشده سیستم مرتفع گردد و نیز قابلیت های سیستم که تولید آنها به تعویق انداخته شده است ، نهایی گردد.
- ✘ این فاز عمدتاً با تولید نسخه بتا (Beta) آغاز می گردد .
- ✘ در پایان این فاز بررسی می شود که آیا همه جوانب سیستم دیده شده است یا باید چرخه ساخت جدیدی را آغاز نمود.

✘ فاز انتقال زمانی آغاز می شود که زیربنای پروژه آنقدر محکم شده باشد که بتوان آنرا برای استفاده کاربر گسترش داد .

- + تست نسخه بتا از لحاظ مطابقت با انتظارات کاربران.
- + تست نسخه بتا همراه با عملکرد موازی سیستم جدید با سیستم قدیمی.
- + تبدیل پایگاه داده.
- + آموزش کاربران و نگهدارندگان سیستم.
- + بازاریابی، توزیع و فروش.

✘ برای برخی نرم افزارها نقطه پایان این چرخه همزمان با نقطه شروع چرخه دیگری است که منجر به تولید نسخه بعدی محصول مورد نظر می شود.