



مقدمه‌ای بر RUP

سید مصطفی حسینی

smhoseyni@gmail.com

فهرست مطالب

❧ RUP چیست؟

❧ مزایای RUP

❧ استفاده از بهترین رویه‌ها

❧ معماری RUP

❧ فازهای RUP

❧ روندهای کاری اصلی

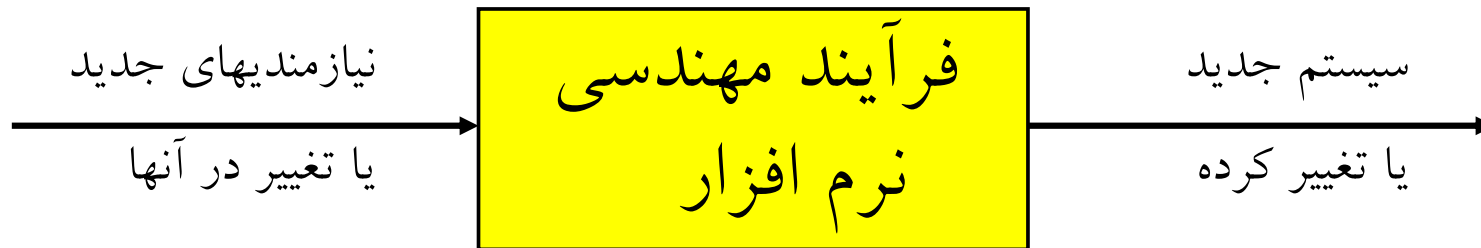
❧ ابزارهای Rational

❧ مطالب جلسات آتی

فرآیند چیست؟

☘ 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.



استانداردها و متدولوژی‌های فرآیند تولید نرم‌افزار

متدولوژی‌های آبخاری (SSADM) ⚡

استانداردهای وزارت دفاع آمریکا: ⚡

MIL-STD-498 ,MIL-STD-7935 ,DoD-STD-2167A ☑

NASA SEL ☑

ISO/IEC 12207 ⚡

IEEE/EIA12207 ☑

روش‌های سریع و چابک (XP, Agile Method, SCRUM) ⚡

روش‌های خاص: PMBOK ,Trillium ⚡

متدولوژی RUP ⚡

استاندارد های ارزیابی کیفیت فرآیند تولید نرم افزار

ISO/IEC 15504 (SPICE) ⌘

SEI - CMM ⌘

CMMI ☑

PSP ☑

BOOTSTRAP ⌘

ISO 9001 ⌘

ISO 9000-3 ☑

Tick IT ☑

RUP چیست؟

⌘ یک فرآیند مهندسی نرم افزار است.

⌘ یک راهکار قانونمند برای انتساب وظایف و مسئولیتها

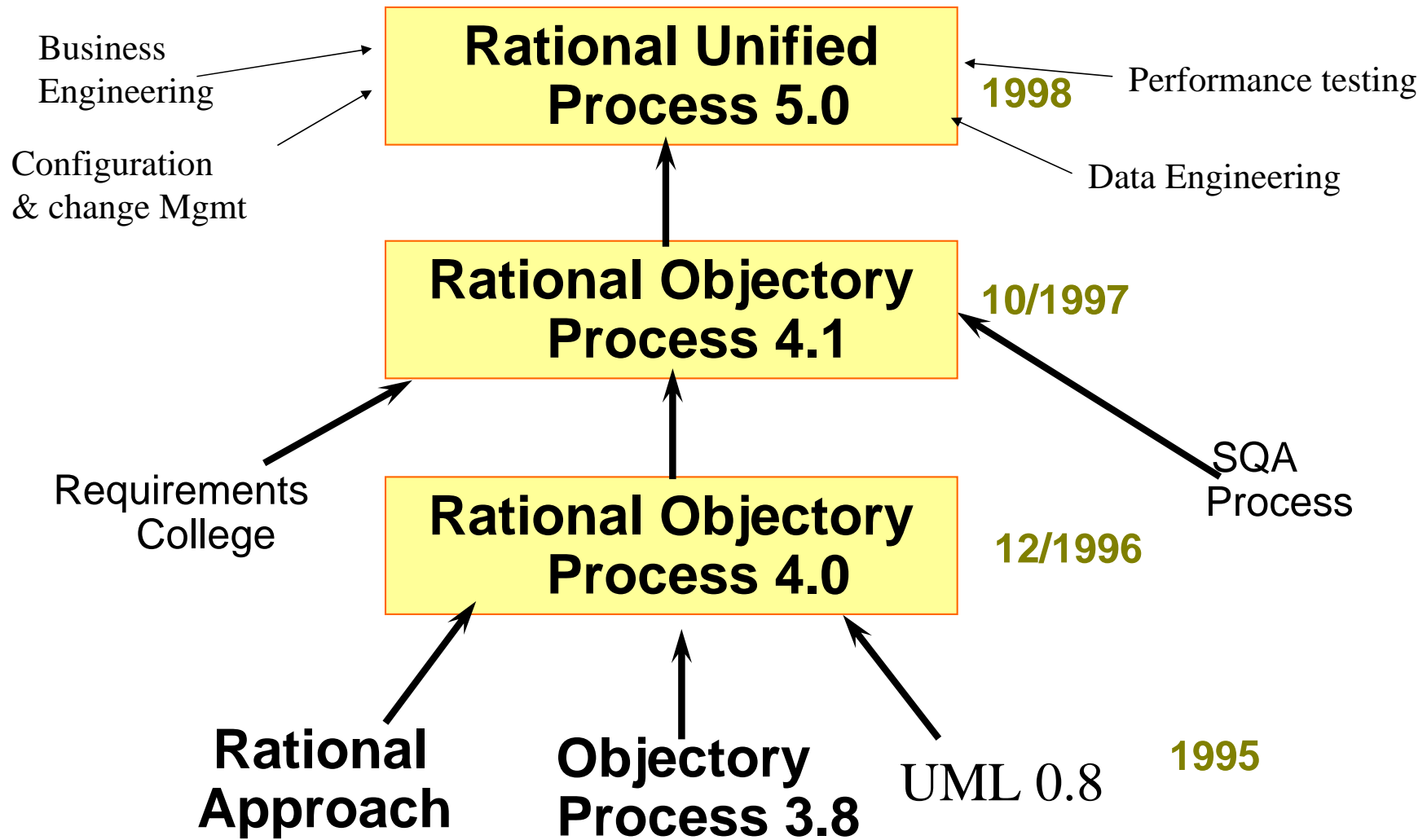
⌘ اهداف:

☑ تضمین کیفیت نرم افزار

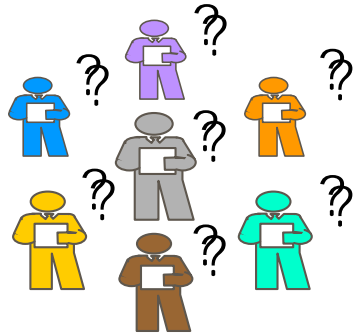
☑ برآورده شدن نیازهای کاربر

☑ قابل تخمین بودن زمان و هزینه تولید

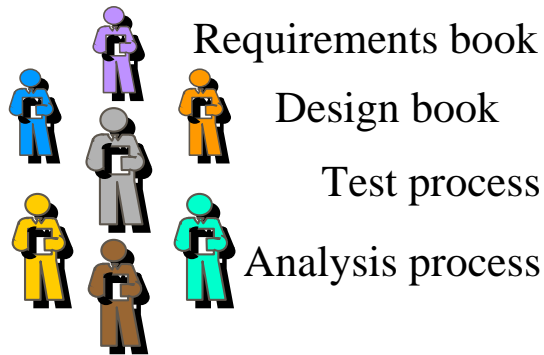
تاریخچه



مشکلات موجود

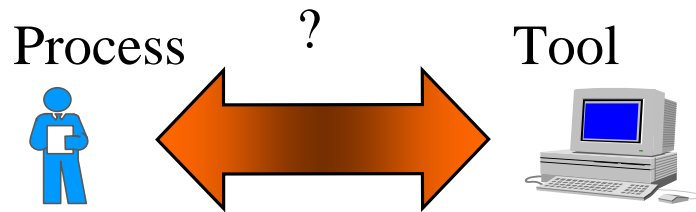


❧ عدم پیروی از فرایند مشخص، یا پیروی از فرآیندهای خودساخته



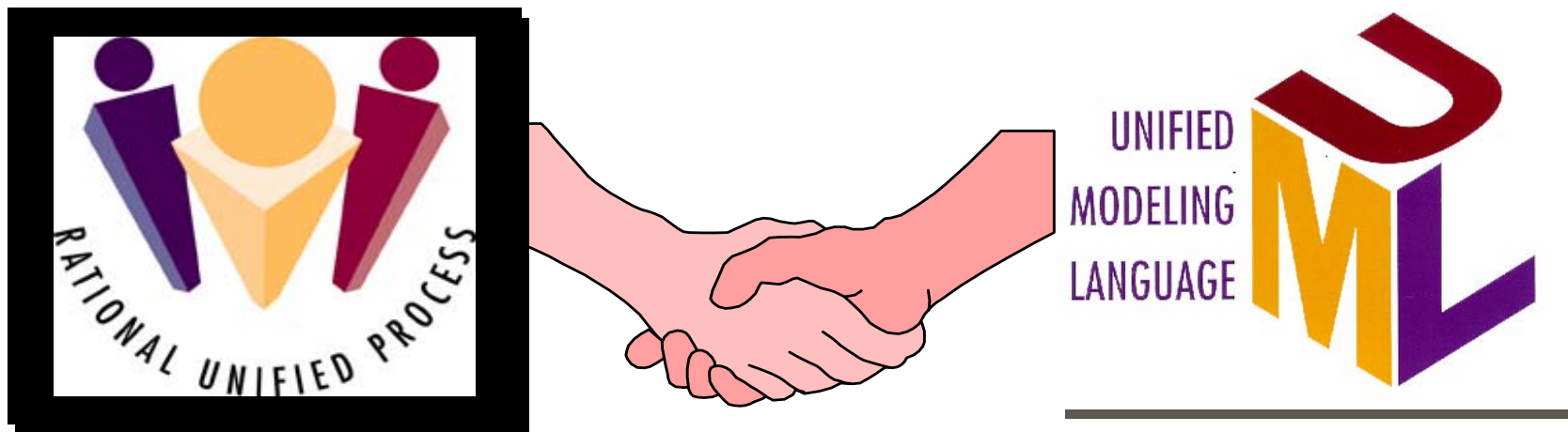
❧ استفاده از فرآیندها و زبانهای متفاوت

❧ عدم پشتیبانی خوب (یا درست) توسط ابزار



راه حل های RUP

- ❖ بهترین بخش های فرایندهای مختلف را در یک فرایند یکپارچه سازگار مجتمع کرده است
- ❖ تولید و نگهداری مدل بجای مستندات کاغذی
- ❖ استفاده کارا و مفید از UML
- ❖ وجود ابزار مناسب برای پوشش کل فرآیند
- ❖ بر بازه گسترده ای از برنامه های کاربردی قابل اعمال است



افزایش بهره‌وری تیم

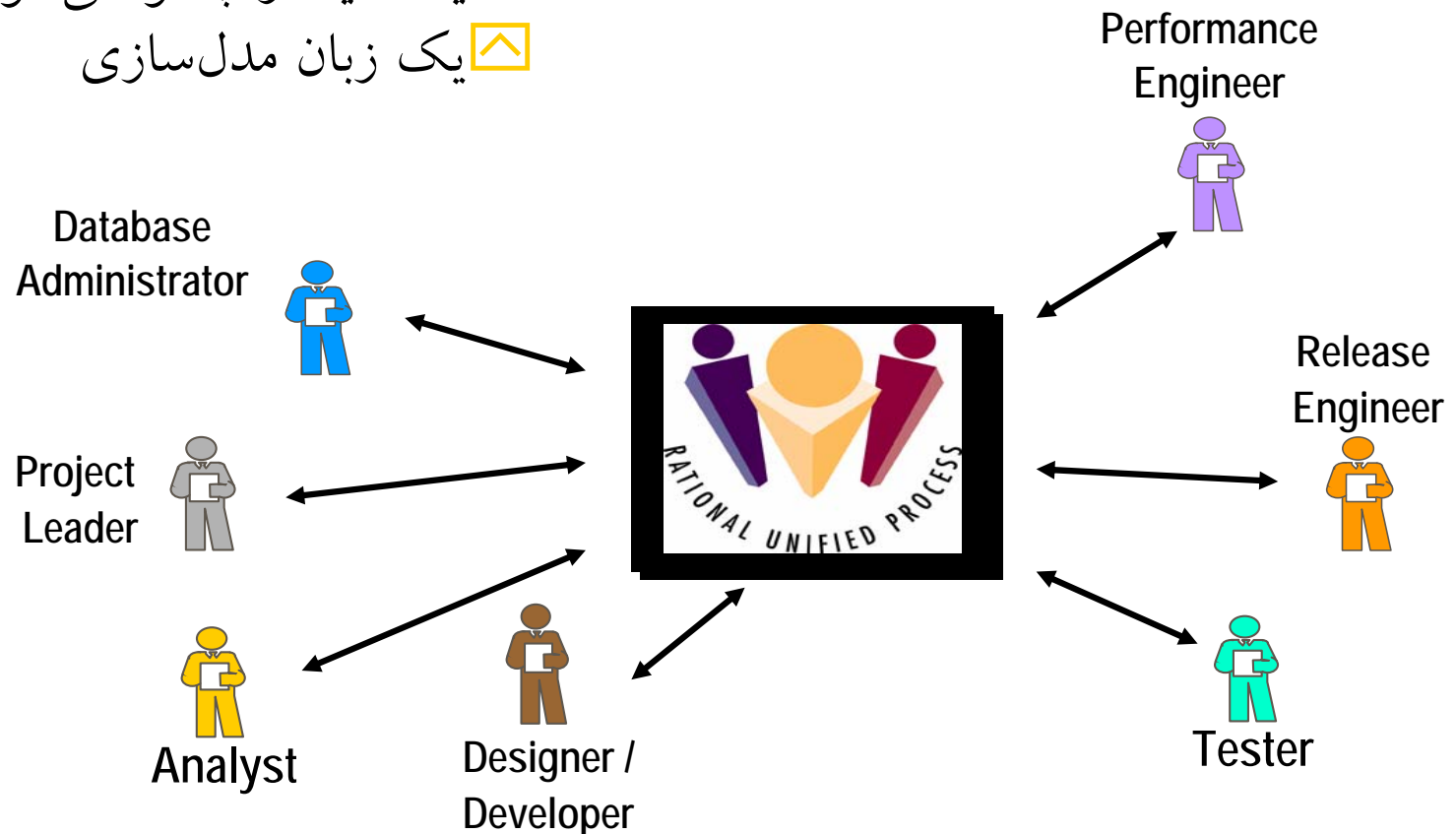
استفاده مشترک اعضا از :

یک پایگاه دانش

یک فرایند

یک دید از چگونگی تولید نرم‌افزار

یک زبان مدل‌سازی



استفاده از بهترین رویه‌ها

1. تولید تکراری و تکاملی
2. مدیریت نیازمندی‌ها
3. استفاده از معماری مبتنی بر اجزاء
4. مدل‌سازی بصری نرم‌افزار
5. تضمین کیفیت نرم‌افزار
6. کنترل تغییرات نرم‌افزار

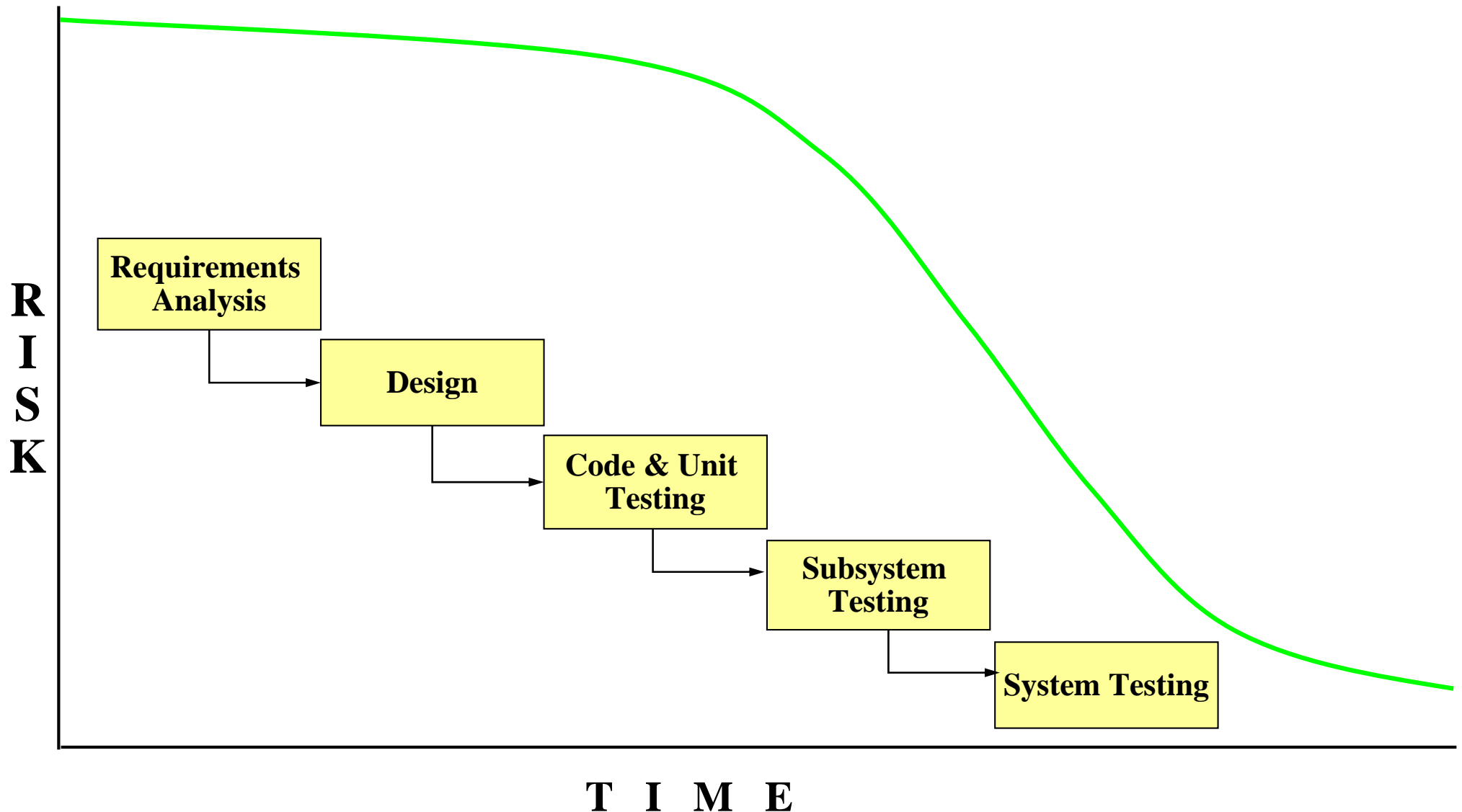
۱- تولید نرم افزار بصورت تکراری و تکاملی

☞ چنانچه در مراحل ابتدایی طراحی صورت گیرد، ممکن است ندیدن برخی از نیازها، طراحی انجام گرفته را با اشکال مواجه کند.

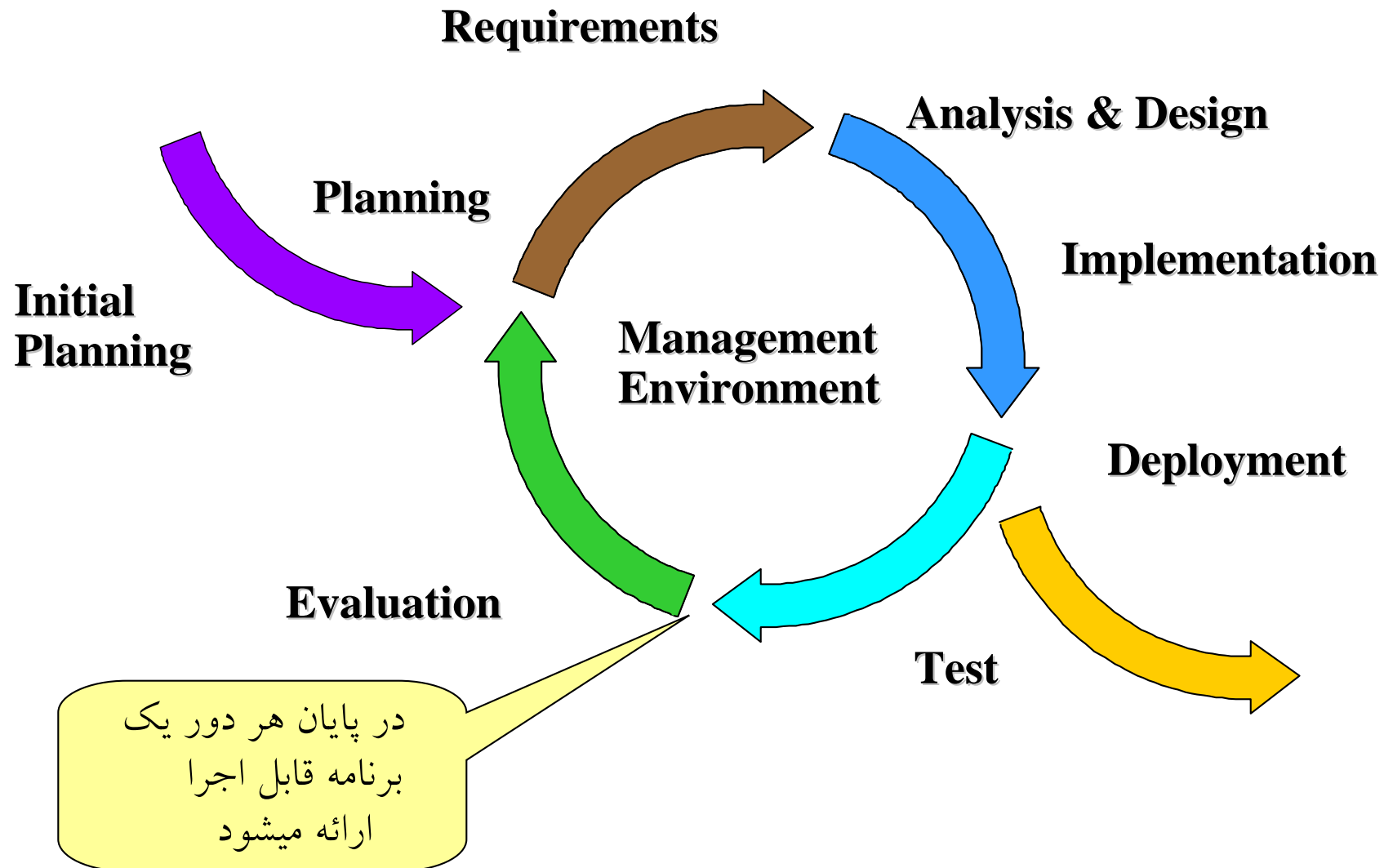
☞ به تأخیر افتادن مشاهده اشکلات طراحی موجب به هدر رفتن منابع و/یا لغو پروژه می شود.

☞ زمان و هزینه‌ای که برای پیاده‌سازی یک طراحی غلط مصرف شده است، غیر قابل بازگشت است.

نمودار ریسک تولید آبخاری



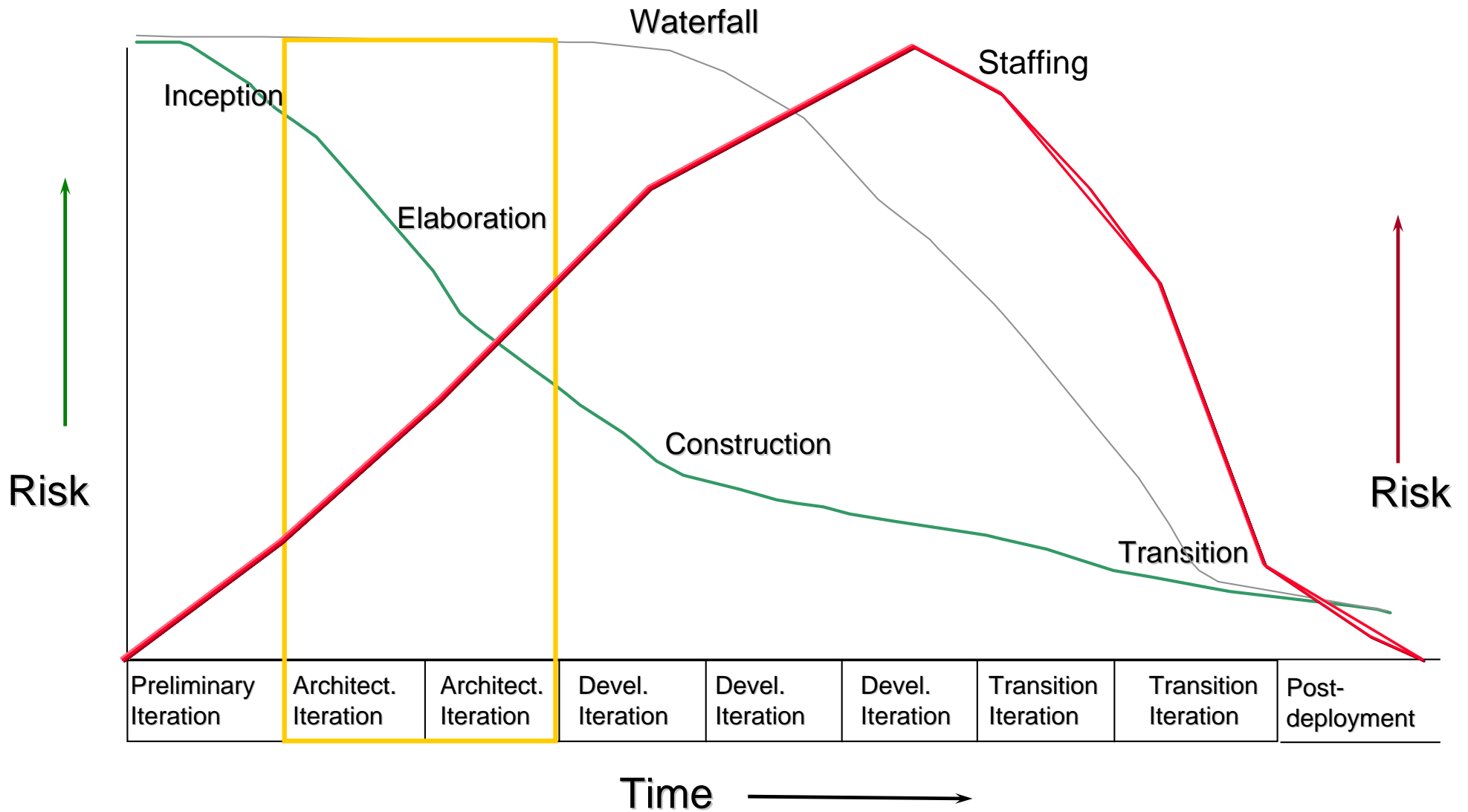
تولید تکراری



ویژگیهای راهکار تکراری و تکاملی

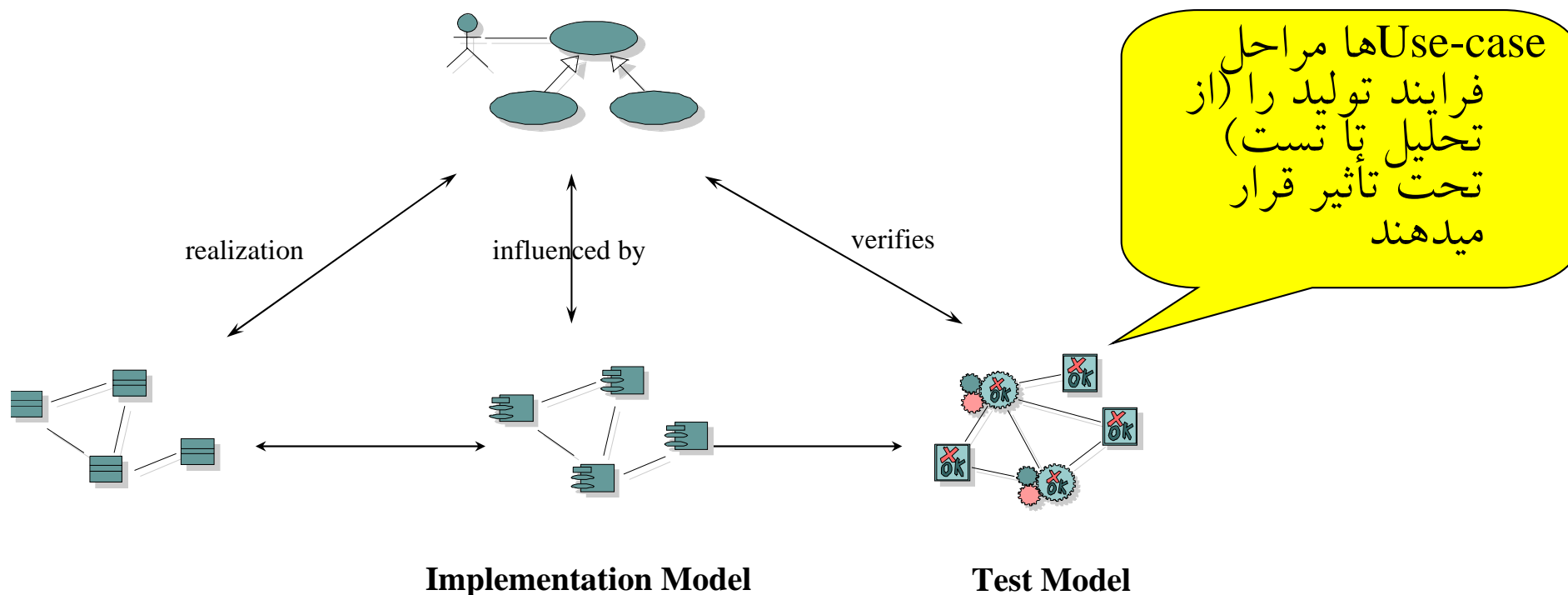
- ❧ رفع ریسک‌های بحرانی پیش از سرمایه‌گذاری‌های بزرگ
- ❧ گرفتن بازخورد کاربر، پس از طی دوره‌های اولیه
- ❧ تست و مجتمع‌سازی مداوم
- ❧ اندازه‌گیری پیشرفت با استفاده از محصول، بجای مستندات یا تخمین‌های مهندسی
- ❧ امکان تحویل برخی پیاده‌سازی‌های جزئی

نمودار ریسک تولید تکراری و تکاملی



۲- مدیریت نیازمندی‌ها

- استخراج، سازماندهی و مستند کردن نیازمندی‌های کاربردی و محدودیت‌ها
- ردیابی و مستند کردن چگونگی تصمیم‌گیری‌ها
- کشف نیازمندی‌های کاری و ارتباطات آنها به سادگی با استفاده از مدل Use-case
- Use-case ها ابزارهای مهمی برای برنامه‌ریزی هستند

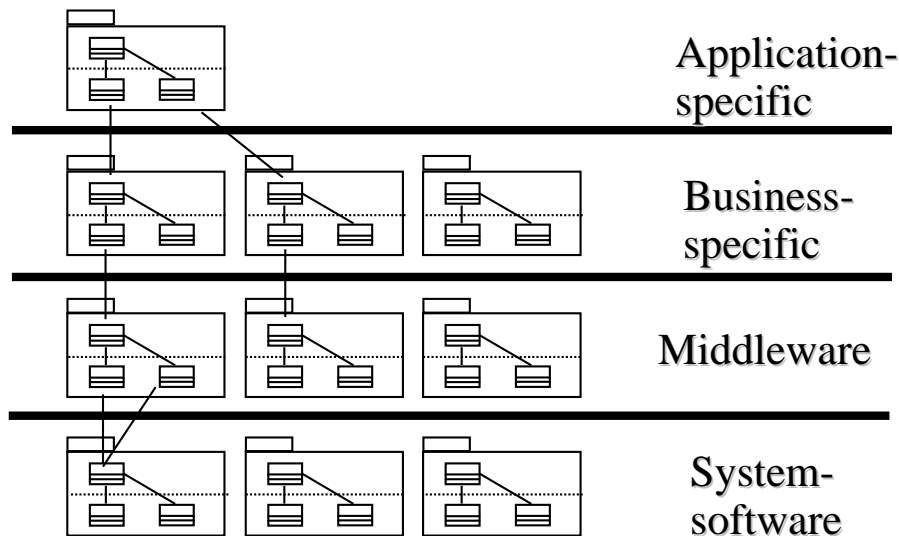


۳- بکارگیری معماری مبتنی بر اجزاء (Component-based)

یک راهکار قانونمند برای تعریف معماری خوب
انعطاف پذیری در برابر تغییرات با استفاده از واسط‌های کاری مناسب

استفاده از اجزاء و مهندسی معکوس اجزاء
مشتق شده از Use-case های با درجه اهمیت بالا

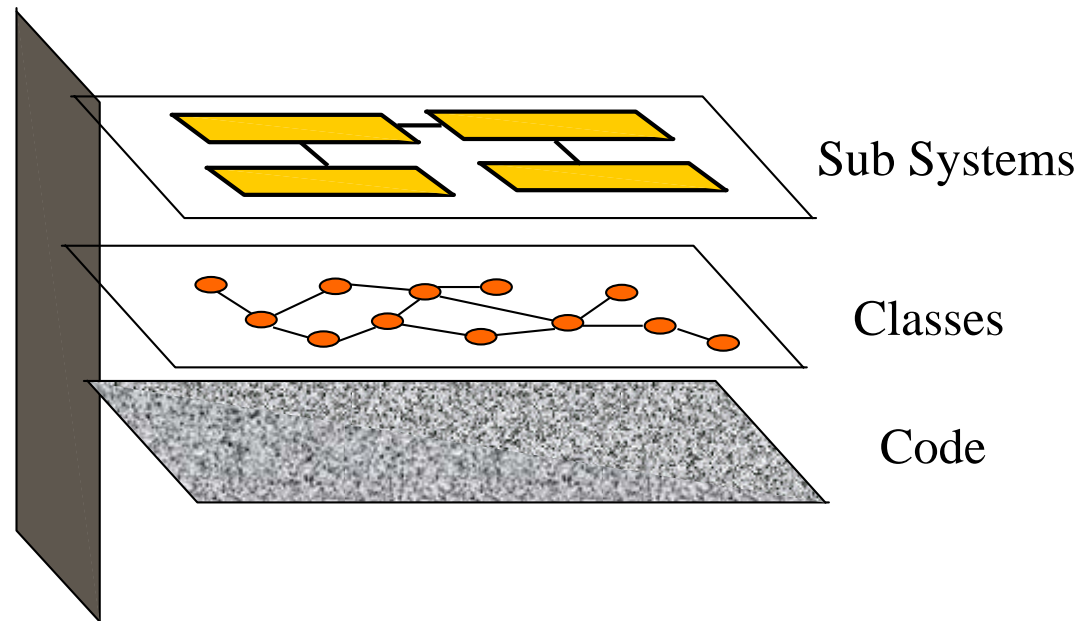
قابل فهم بصورت شهودی



معماری لایه لایه و
مبتنی بر اجزاء

۴- مدل سازی بصری نرم افزار

- نمایش ساختار و رفتار معماری ها و اجزاء
- نشان دادن چگونگی تطابق اجزاء سیستم با یکدیگر
- حفظ سازگاری بین طراحی و پیاده سازی
- ترویج ارتباطات واضح و غیر گنگ



مدل سازی بصری
سطوح تجرد را
افزایش میدهد

۵- تضمین کیفیت نرم افزار

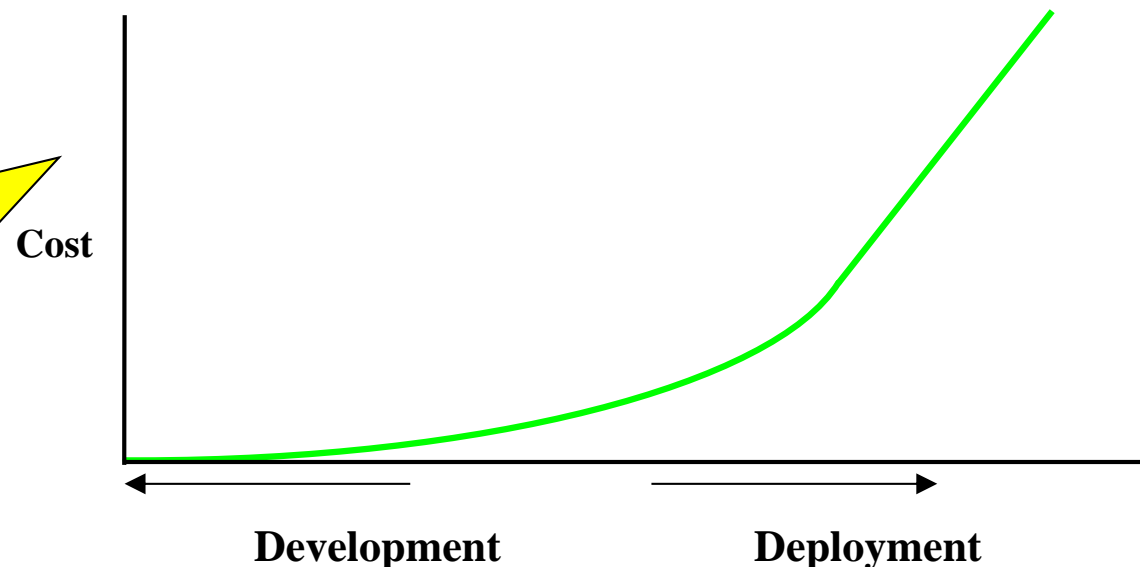
☞ تولید آزمون‌هایی برای هر سناریوی کلیدی تا اطمینان حاصل شود که تمام نیازمندی‌ها بدرستی پیاده‌سازی شده است.

☞ ارزیابی کارایی و قابلیت اطمینان

☞ تضمین قابلیت اطمینان - نشت حافظه و گلوگاه‌ها

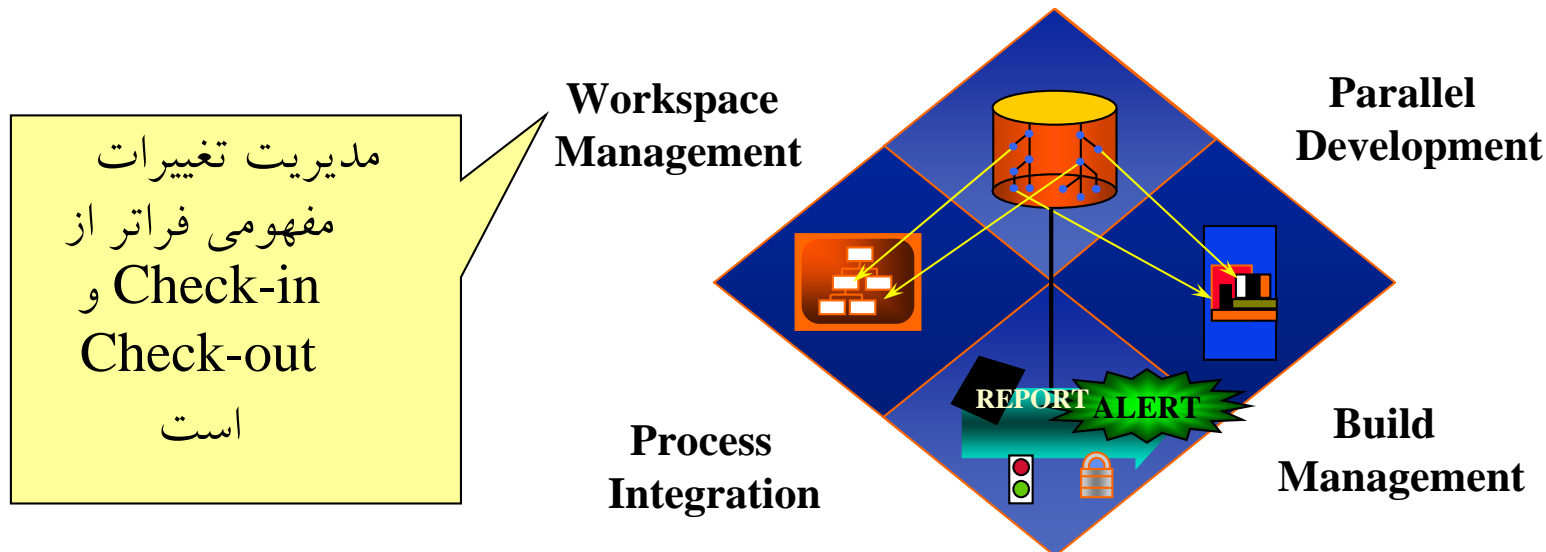
☞ تست و آزمایش هر تکرار - تست خودکار

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



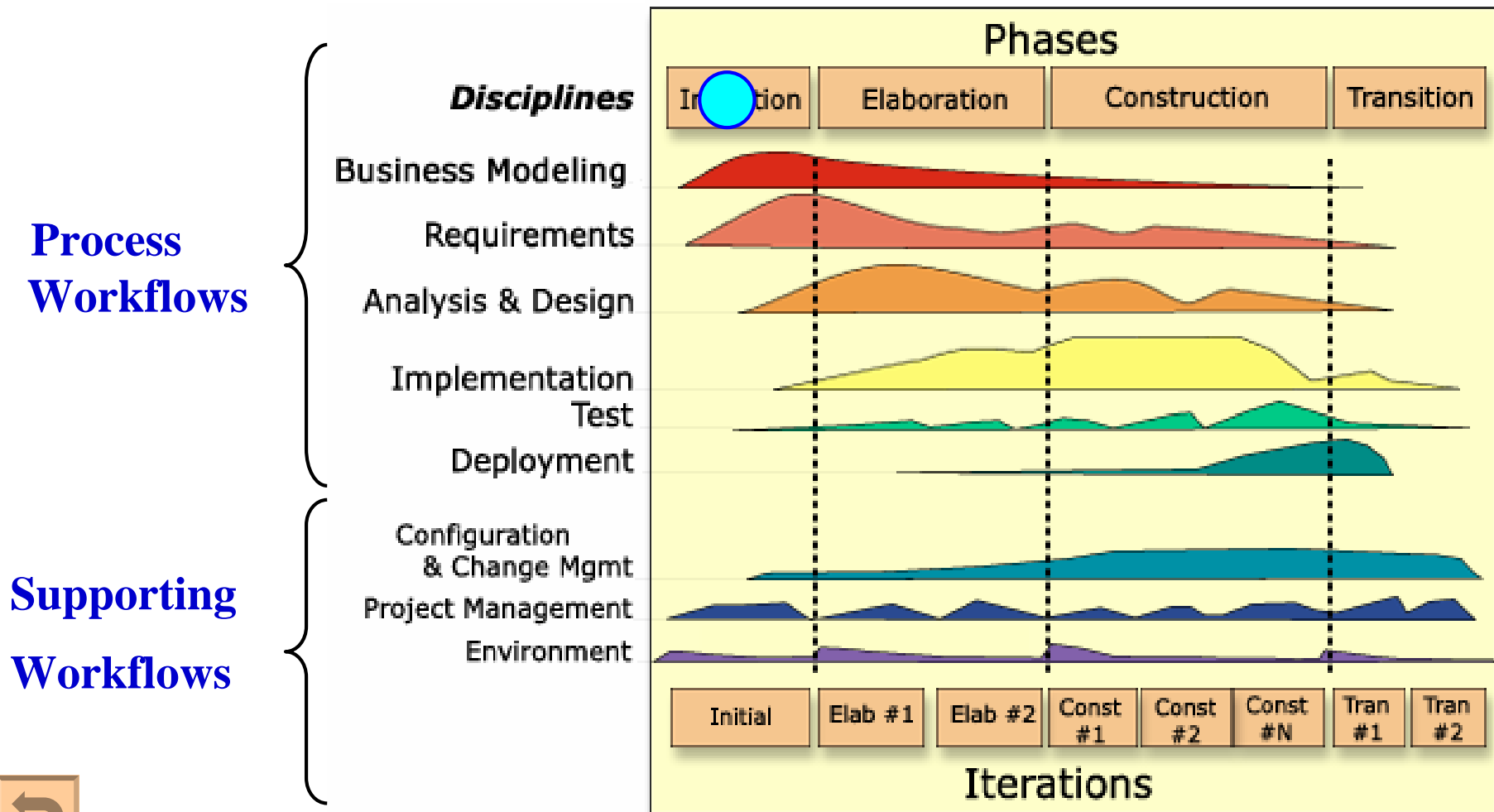
۶- کنترل تغییرات نرم افزار

- ☞ کنترل، ردیابی و نظارت بر تغییرات به منظور امکان بخشیدن تولید تکاملی
- ☞ بنا کردن فضای ایمن برای هر تولیدکننده
- ☒ جلوگیری از اثر تغییرات در یک فضای کاری بر روی سایر فضاهای کاری
- ☒ کنترل تمام محصولات نرم افزاری: مدل، کد، مستندات و غیره.
- ☞ مجتمع سازی خودکار و مدیریت ساخت

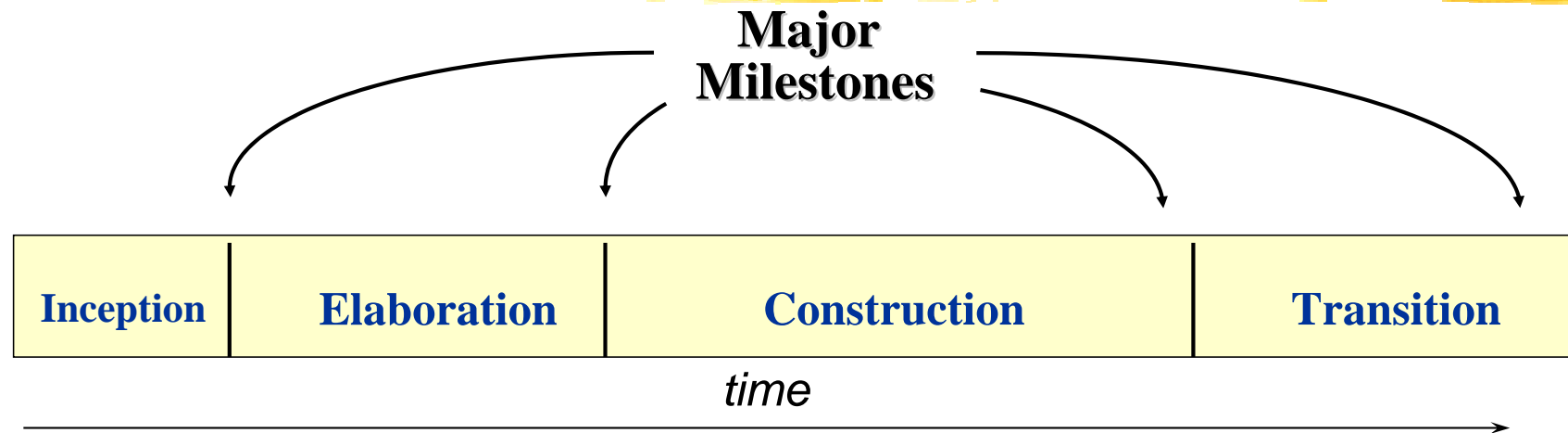


معماری فرآیند

Time



فازهای فرآیند



RUP دارای ۴ فاز است : ❧

Inception: تعریف حوزه مسأله ☒

Elaboration: برنامه ریزی پروژه، مشخص کردن ویژگی‌ها، پایه‌گذاری معماری ☒

Construction: ساختن محصول ☒

Transition: انتقال محصول به محیط کاربر ☒

فاز Inception (شروع)

☞ هدف :

- ☑ دستیابی به توافق میان کلیه ذینفعان بر روی اهداف چرخه‌ی حیات پروژه
- ☑ بدست آوردن محدوده نرم‌افزاری پروژه و محدودیت‌های آن (Visoin)
- ☑ مشخص کردن سناریوهای اساسی سیستم (Use-Case)
- ☑ برآورد هزینه و زمان کلی برای کل پروژه

☞ خروجی :

- ☑ یک دید کلی از نیازمندی‌های پروژه (Vision)
- ☑ مدل Use-case ابتدایی
- ☑ یک business case ابتدایی که شامل ضوابط و شرایط موفقیت و پیش‌بینی‌های مالی است.
- ☑ یک برآورد ابتدایی از ریسک‌ها
- ☑ تخمینی از منابع مورد نیاز
- ☑ یک طرح برای پروژه که فازها و تکرارها را نمایش دهد

فاز Elaboration (جزئیات)

اهداف

- تحلیل دامنه مسأله
- بنا کردن شالوده معماری استوار
- حذف ریسک‌های عمده پروژه
- ایجاد یک طرح فراگیر که نشان دهد چگونه پروژه کامل خواهد شد

خروجی :

- تکمیل مدل use-case تا ۸۰٪
- یک معماری قابل اجرا به همراه مستندات
- Business case پالایش شده که شامل برآورد ریسک‌های پالایش شده است
- یک طرح توسعه برای سرتاسر پروژه

فاز Construction (ساخت)

اهداف:

تولید تکاملی یک محصول نرم‌افزاری کامل که قابل انتقال به محیط کاربر باشد.

خروجی:

یک مدل طراحی و مدل Use-Case کامل

نسخه قابل اجرای نرم‌افزار که تمام نیازهای کارکردی را برآورده میکند

مستندات کاربر

فاز Transition (انتقال)

اهداف:

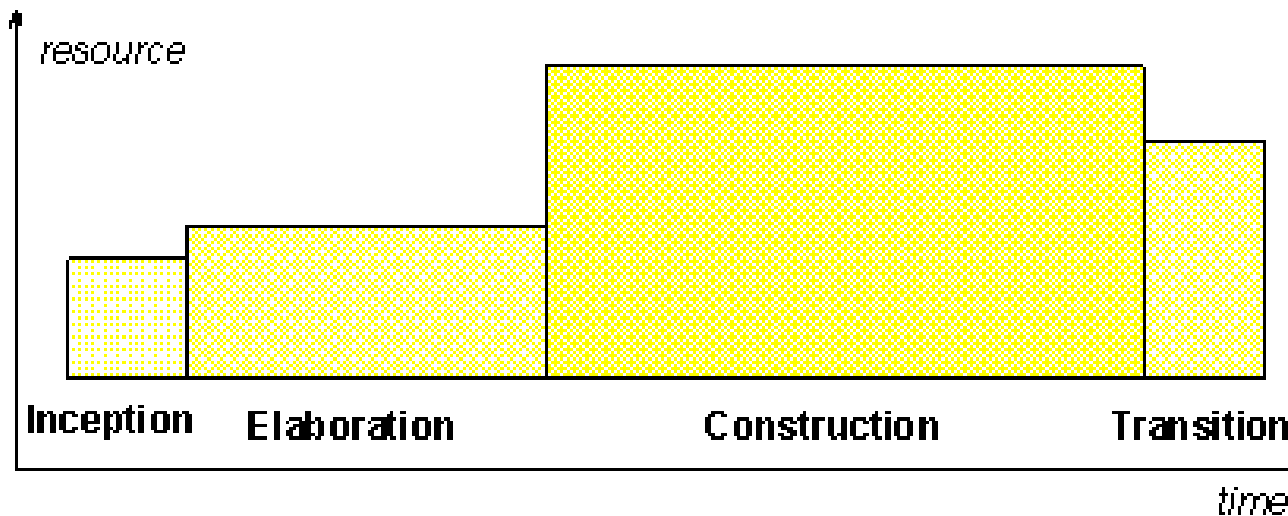
انتقال محصول نرم‌افزاری به محیط کاربر

خروجی:

- انتظار کاربران در برابر انتظارات کاربران
- Beta testing به منظور ارزیابی سیستم در برابر انتظارات کاربران
- اجرای موازی همزمان با سیستم موجود که جایگزین آن خواهد شد
- تبدیل کردن پایگاه داده‌های قابل استفاده
- آموزش کاربران و نگهداران سیستم
- آماده کردن محصول برای بازاریابی و فروش
- مدل‌های سیستم به‌روزشده

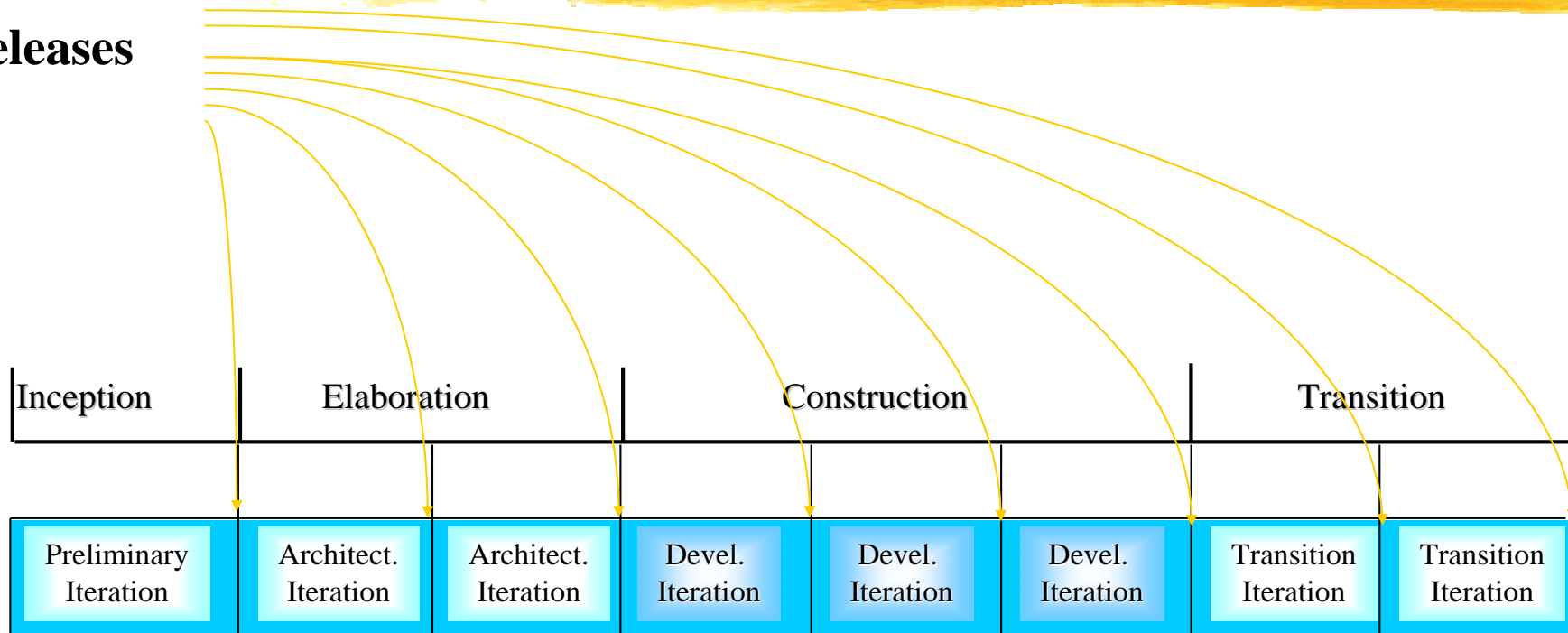
بکارگیری منابع

	Inception	Elaboration	Construction	Transition
Effort	~5 %	20 %	65 %	10%
Schedule	10 %	30 %	50 %	10%



تکرارها و فازها

Releases

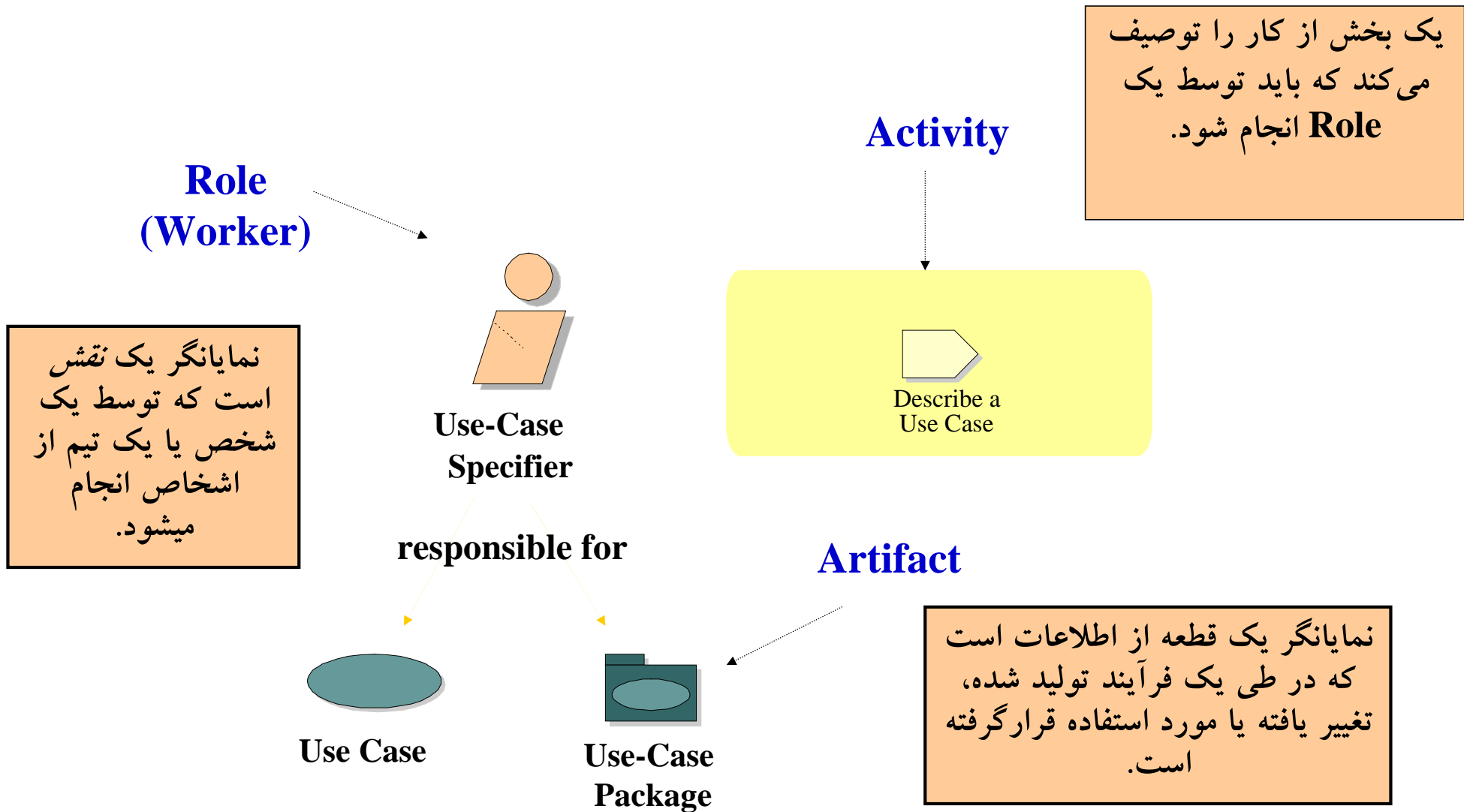


✂ هر تکرار یک دنباله جداگانه از اعمال به همراه برنامه ساخت و معیارهای ارزیابی است که یک نسخه قابل اجرا در انتهای آن بدست می آید.

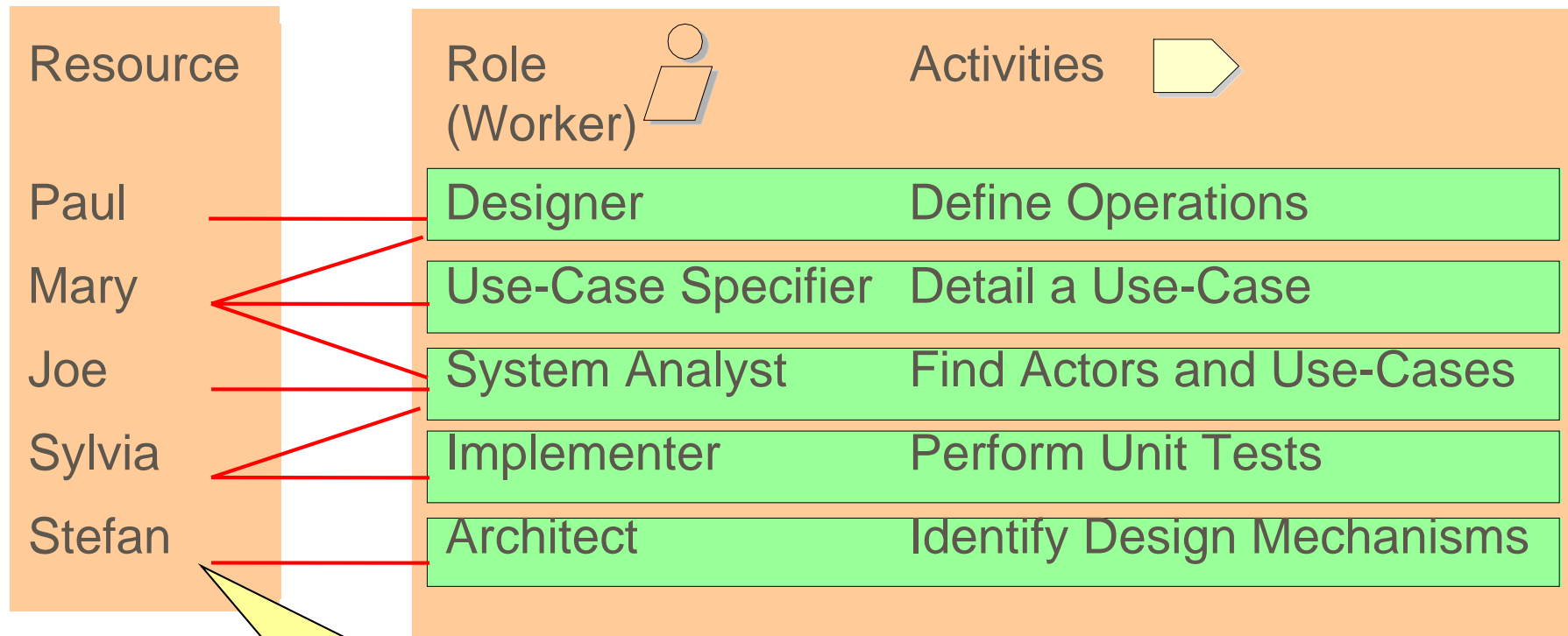
✂ منافع راهکار تکراری :

☑ کاهش زود هنگام ریسک ها، قابل مدیریت بودن تغییرات، استفاده مجدد در سطوح بالاتر، آموزش تیم در طول زمان، کیفیت کلی بهتر

نوشتار فرآیند



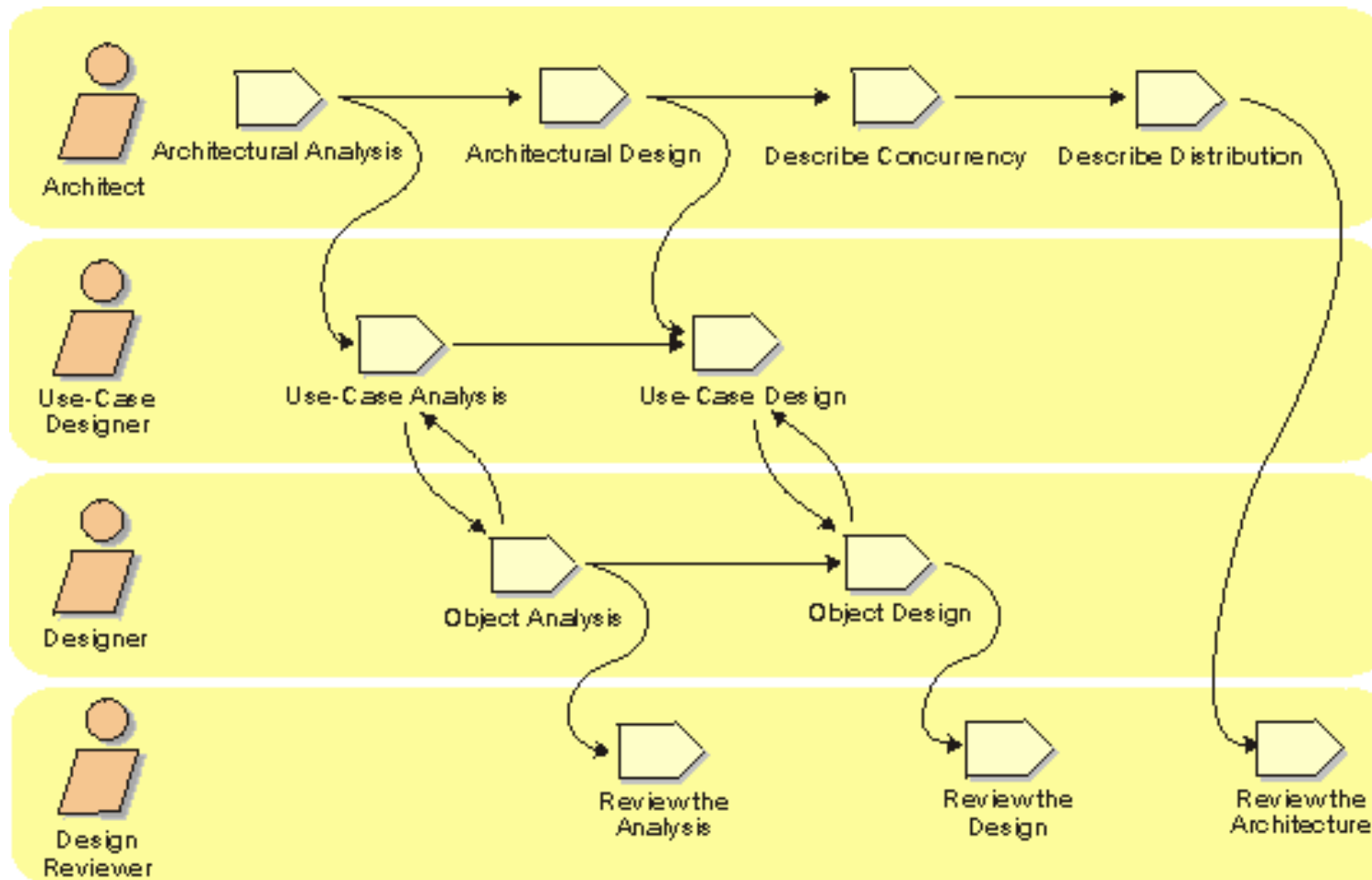
از Role ها برای برنامه ریزی منابع استفاده میشود



هر شخص در پروژه به
یک یا چند Role
منتسب می شود

روند کاری (Workflow)

یک دنباله از فعالیت‌ها که منجر به تولید نتیجه‌ای با ارزش قابل ملاحظه خواهد شد.



روندهای کاری اصلی (Core Workflows)

☞ روندهای کاری مهندسی

1. مدلسازی سیستم موجود (Business Modeling)
2. استخراج نیازمندی‌ها (Requirements)
3. تحلیل و طراحی (Analysis & Design)
4. پیاده‌سازی (Implementation)
5. آزمون (Test)
6. استقرار (Deployment)

☒ روندهای کاری پشتیبانی

1. مدیریت پروژه (Project Management)
2. مدیریت پیکربندی و تغییرات (Configuration & Change Mgmt.)
3. مدیریت محیط (Environment)



مدل سازی کسب و کار (Business Modeling)

اهداف:

- مستند سازی فرآیندهای سیستم موجود با استفاده از Business use case
- فهم مشترک در میان تمام منتفعان سیستم در سازمان
- تحلیل business use case ها به منظور کشف اینکه چگونه فرآیندها در سیستم موجود انجام می شوند.

فرآورده‌ها:

- Business rules
- Business vision
- Business glossary
- Business architecture
- Business object model
- Business use-case model

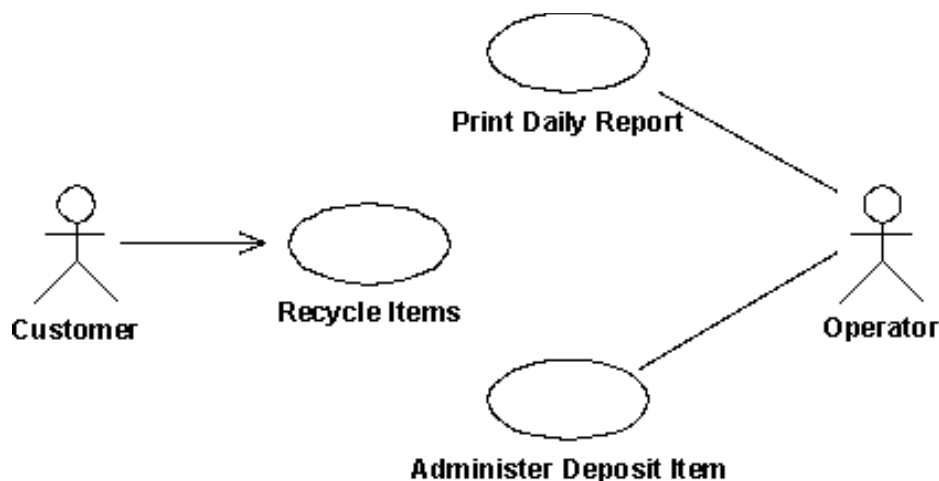
نیازمندی‌ها (Requirements)

☞ اهداف :

☑ یک توصیف از آنچه که سیستم باید انجام دهد بطوریکه مورد توافق مشتری و تولید کننده باشد

☑ استخراج، سازماندهی و مستندسازی نیازهای وظیفه‌مندی و محدودیت‌ها

☞ فرآورده‌ها :



Glossary ☑

Use-case model ☑

Vision document ☑

Stakeholder requests ☑

Supplementary specification ☑

User-interface prototype ☑

تحليل و طراحی (Analysis & Design)

اهداف:

توصیف اینکه سیستم چگونه پیاده‌سازی خواهد شد.

فرآورده‌ها:

Data model

Design model

Software architecture document

پیاده سازی (Implementation)

اهداف:

- تعریف چگونگی سازماندهی کد در قالب زیرسیستمها و لایهها
- پیاده سازی کلاسها و اشیاء در قالب component ها
(فایل های source ، binary ، executable ، ...)
- تست واحد اجزاء پیاده سازی شده (unit test)
- مجتمع سازی اجزائی که بوسیله تولیدکنندگان مختلف تولید شده است در قالب یک برنامه قابل اجرا

فرآوردهها:

Build

Implementation model

آزمون (Test)

اهداف :

- ارزیابی صحت تعامل میان اشیاء
- ارزیابی صحت مجتمع‌سازی تمام اجزاء نرم‌افزار
- ارزیابی اینکه تمام نیازمندی‌ها بدرستی پیاده‌سازی شده است
- اطمینان از تشخیص عیوب نرم‌افزار و رسیدگی به آنها پیش از تحویل

فرآورده‌ها :

- Test Plan
- Test Results
- Test-Ideas List
- Test Evaluation Summary

استقرار (Deployment)

☞ اهداف :

☑ ارائه موفقیت آمیز نسخه نهایی و تحویل نرم افزار به کاربران آن

☞ فعالیت ها :

☑ تولید نسخه خارجی نرم افزار

☑ بسته بندی نرم افزار

☑ توزیع نرم افزار

☑ نصب نرم افزار

☑ فراهم کردن کمک و راهنمایی برای کاربران

☞ فرآورده ها :

☑ Deployment Plan

☑ End-user support material

مدیریت پروژه (Project Management)

اهداف: ☘

برقراری تعادل میان اهداف رقیب ☒

مدیریت ریسکها ☒

غلبه بر محدودیتها و ارائه محصولی که نیازهای مشتری و کاربران را برآورده کند ☒

فرآوردهها: ☘

Business case ☒

Issue list ☒

Iteration plan ☒

Iteration assessment ☒

Risk list ☒

Software development plan ☒

Status assessment ☒

مدیریت پیکربندی و تغییرات (Configuration & Change Mgmt.)

اهداف:

توصیف چگونگی کنترل محصولات زیادی که اعضای تیم پروژه تولید

کرده‌اند و رفع مشکلاتی همچون:

به روزآوری همزمان

عدم خبردهی مناسب تغییرات

مدیریت نسخه‌های چندگانه

فرآورده‌ها:

Change requests

Project repository

Workspace

محیط تولید (Environment)

اهداف:

فراهم کردن سازمان تولید نرم افزار

فراهم کردن محیط تولید نرم افزار (فرآیند و ابزارها)

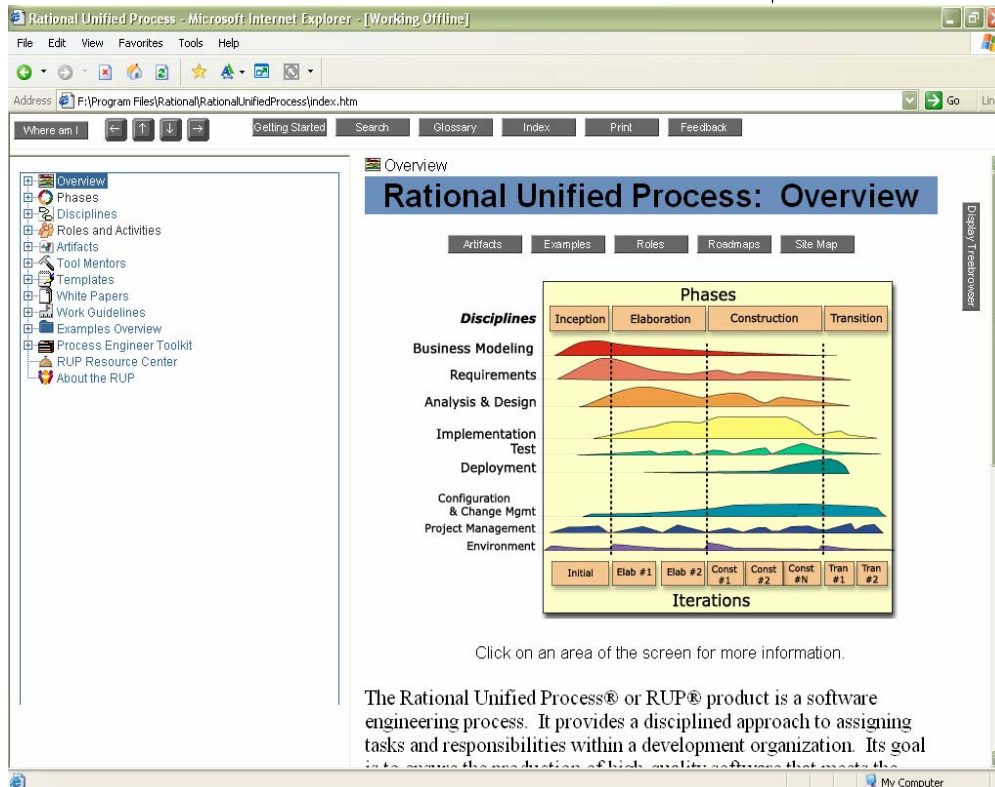
خروجی ها:

Development case

Development infrastructure

محصول RUP

⌘ یک پایگاه دانش مبتنی بر web و قابل جستجو
⌘ قابل استفاده برای تمام مراحل فرآیند تولید نرم افزار
⌘ شامل:



⌘ راهنمایی هایی برای تمام اعضای تیم

⌘ خود آموز ابزار

⌘ مثالها و قالبهایی از Rational Rose

⌘ قالبهای MS-World و SoDA

ابزارهای پشتیبانی کننده RUP (محصولات IBM Rational)

- Rose ❌ (مدل سازی بصری اجزاء برطبق UML و تولید کد)
- XDE ❌ (مدل سازی بصری اجزاء و تولید کد خودکار برای .NET و JAVA)
- RequisitePro ❌ (سازماندهی، اولویت بخشی، ردیابی و کنترل تغییرات نیازمندی ها)
- SoDA ❌ (قالب های MS-Word برای تولید مستندات)
- PurifyPlus ❌ (خطاهای زمان اجرا، نقاطی از کد که تست نشده اند، ارزیابی کارایی)
- Robot ❌ (ایجاد، تغییر و اجرای آزمون های عملکردی بر روی برنامه)
- TestFactory ❌ (تولید خودکار test script های جامع و فراگیر)
- Functional Tester ❌ (آزمون خودکار عملکردی برای .NET، Java و Web)
- ClearQuest ❌ (ردیابی نقایص و مدیریت درخواست های تغییر)
- ClearCase ❌ (مدیریت پیکربندی)
- Rational Software Architect (RSA) ❌ (ابزار پیشرفته و جامع تولید)

مطالب جلسات آتی - ۱

☞ معرفی دیسیپلین Business Modeling، نقش‌ها، فرآیند، فعالیت‌ها، مفاهیم و فرآورده‌ها

☞ آشنایی با مفاهیم (Balanced Scorecard (BSC، نمودار Fishborn، نمودار Pareto، Activity-Based Costing

☞ آشنایی با دست‌آوردها: Business Vision، Business Goal، Business Rule

☞ آشنایی با Business Use-Case Modle، مفاهیم و قواعد، روابط (Association، Generalization، Extend، Include)

☞ مدل‌سازی Business Actor و Business Use-Case در Rational Rose (یا XDE) - مثال و تمرین

☞ آشنایی با Business Analysis Model (Business Worker، Business Entity، Business System)

☞ مدل‌سازی Business Analysis Model در Rational Rose (یا XDE) - مثال و تمرین

☞ معرفی دیسیپلین Requirements، نقش‌ها، فرآیند، فعالیت‌ها، مفاهیم و فرآورده‌ها

☞ آشنایی با دست‌آوردها: Vision، Supplementary Requirement

☞ آشنایی با Actor & Use-Case و تشریح مفاهیم آنها با مثال و تمرین

☞ روابط Communication-Association، Include، Extend، Generalization

☞ آشنایی با ابزار RequisitePro (روابط ردگیری، صفات نیازمندی‌ها، کنترل تغییرات، ارتباط با MS-Word)

☞ مدل‌سازی Actor و Use-Case در Rational Rose (یا XDE)

مطالب جلسات آتی - ۲

☞ معرفی دیسیپلین Analysis & Design ، نقش‌ها، فرآیند، فعالیت‌ها، مفاهیم و فرآورده‌ها

☒ آشنایی با Analysis Model (کلاس‌های Entity, Bndry, Control)

☒ تحلیل رفتار، Use-Case Realization

☒ مدل‌سازی کلاس‌ها در Rose (یا XDE) - مثال و تمرین

☒ روابط Aggregation, Association, Composition, Dependency

Generalization و نحوه ترسیم آنها - مثال و تمرین

☒ نمودارهای Sequence (Collaboration), Communication, Statechart و

Activity - مثال و تمرین

☒ معرفی الگوهای طراحی و کاربردهای آنها (GoF) Design Pattern

☒ Composite, Strategy, Decorator, Abstract Factory, Bridge, Iterator, Command و Visitor

☒ استفاده از الگوهای طراحی در Rose (XDE)

مطالب جلسات آتی - ۳

☘ معرفی دیسیپلین Implementation ، نقش‌ها، فعالیت‌ها و فرآورده‌ها

☒ آشنایی با فرآورده‌ها : Component, Build, Integration Build Plan

☒ مدل سازی Implementation Diagram در Rose (XDE) - مثال و تمرین

☒ تولید خودکار کد از روی مدل

☒ تجزیه مدل برای تولید همروند

☘ معرفی دیسیپلین Test ، نقش‌ها، فعالیت‌ها و فرآورده‌ها

☒ آشنایی با فرآورده‌ها : Test Data, Test Suite, Test Case, Test Plan

☒ آشنایی و تمرین با ابزارها Purify, PureCoverage, Quantify, Functional Tester

☘ معرفی دیسیپلین Deployment ، نقش‌ها، فعالیت‌ها و فرآورده‌ها

☘ معرفی دیسیپلین Configuration & Change Mgmt. ، فعالیت‌ها و فرآورده‌ها

☒ آشنایی با ابزار ClearQuest

☘ معرفی دیسیپلین Project Management ، نقش‌ها، فعالیت‌ها و فرآورده‌ها

راجع به ارائه کننده



☞ سید مصطفی حسینی

☞ کارشناس ارشد مهندسی نرم افزار، دانشگاه صنعتی شریف

☞ مهارت‌ها

☞ آموزش متدولوژی RUP، زبان UML و ابزارهای IBM Rational

☞ سفارشی نمودن RUP (RUP Tailoring) با توجه به نیازهای و شرایط

☞ راهبری تیم تولید در استفاده از RUP (Process Engineering)

☞ مشاوره در زمینه RUP، معماری نرم افزار و تکنولوژی‌های نرم افزاری

☞ نظارت بر پروژه‌های نرم افزاری از جهت تطابق با متدولوژی

☞ معماری و طراحی سیستم‌های توزیع شده، رخداده‌ها، پردازشگر Packet و سیستم‌های پیچیده دیگر

☞ برنامه‌نویسی در هسته سیستم عامل لینوکس

☞ آشنا به تکنولوژی‌های J2EE و Microsoft .NET

☞ برنامه نویسی با استفاده از تکنولوژی J2EE و .NET. و مسلط به زبان‌های ++C، C#، Java و Delphi

☞ نحوه تماس

☞ Site : www.smhoseyni.com

☞ E-Mail : smhoseyni@gmail.com

☞ Mobile : 0912 275 37 87