

بسمه تعالی

فصل پنجم

معرفی فرآیند تولید نرم افزار در متدولوژی USDP

مدرس: فریدون شمس

اهداف جلسه

- معرفی فرآیند تولید در متدولوژی **USDP**
- آشنائی با روشهای مدرن توسعه نرم افزار و ویژگیهای آنها
- درک تفاوت بین **USDP** و **RUP**
- معرفی محورهای اصلی **USDP**
- درک محصول بودن **RUP**
- آشنائی با ابعاد **RUP**

فهرست

- مقدمه ای بر فرآیند USDP
- روشهای مدرن توسعه نرم افزار
- محورهای اصلی USDP
- RUP بعنوان یک محصول
- معرفی ابعاد فرآیند RUP

مقدمه‌ای بر فرآیند USDP

یک فرآیند تولید نرم افزار چهار نقش اساسی دارد:

(۱) مشخص نمودن ترتیب فعالیت‌هایی که باید صورت گیرد تا نیازمندی‌های کاربران به یک محصول واقعی تبدیل شوند.

(۲) بیان اینکه چه فرآورده‌هایی باید تولید شوند و در چه زمانی

(۳) تعیین روش اداره وظایف توسعه دهندگان منفرد و تیمی، نقش‌های مورد نیاز در پروژه (سیستم

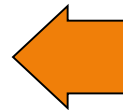
مقدمه‌ای بر فرآیند USDP

(ادامه)

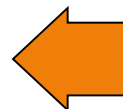
۴) فراهم نمودن معیارهایی (Software Metrics) برای اندازه‌گیری کیفیت محصولات پروژه و روند پیشرفت فعالیت‌های آن

نرم افزارهای مورد نیاز به صورت منظم و قابل پیش بینی تولید قابل تولیدند

موفقیت بستگی به تلاش طاقت فرسای اعضای تیم دارد



به خوبی مستند شده باشد



به خوبی مستند شده نباشد

در فرآیند تولید که

مقدمه‌ای بر فرآیند USDP

(ادامه)

” یکی از مشخصات بارز یک فرآیند تولید خوب استفاده از تجربیات بدست آمده از اجرای پروژه های نرم افزاری موفق است (روشهای مدرن توسعه نرم افزار) “

تکرار و توسعه تدریجی

مدیریت
نیازمندیها

استفاده از
معماری مبتنی
بر مؤلفهها

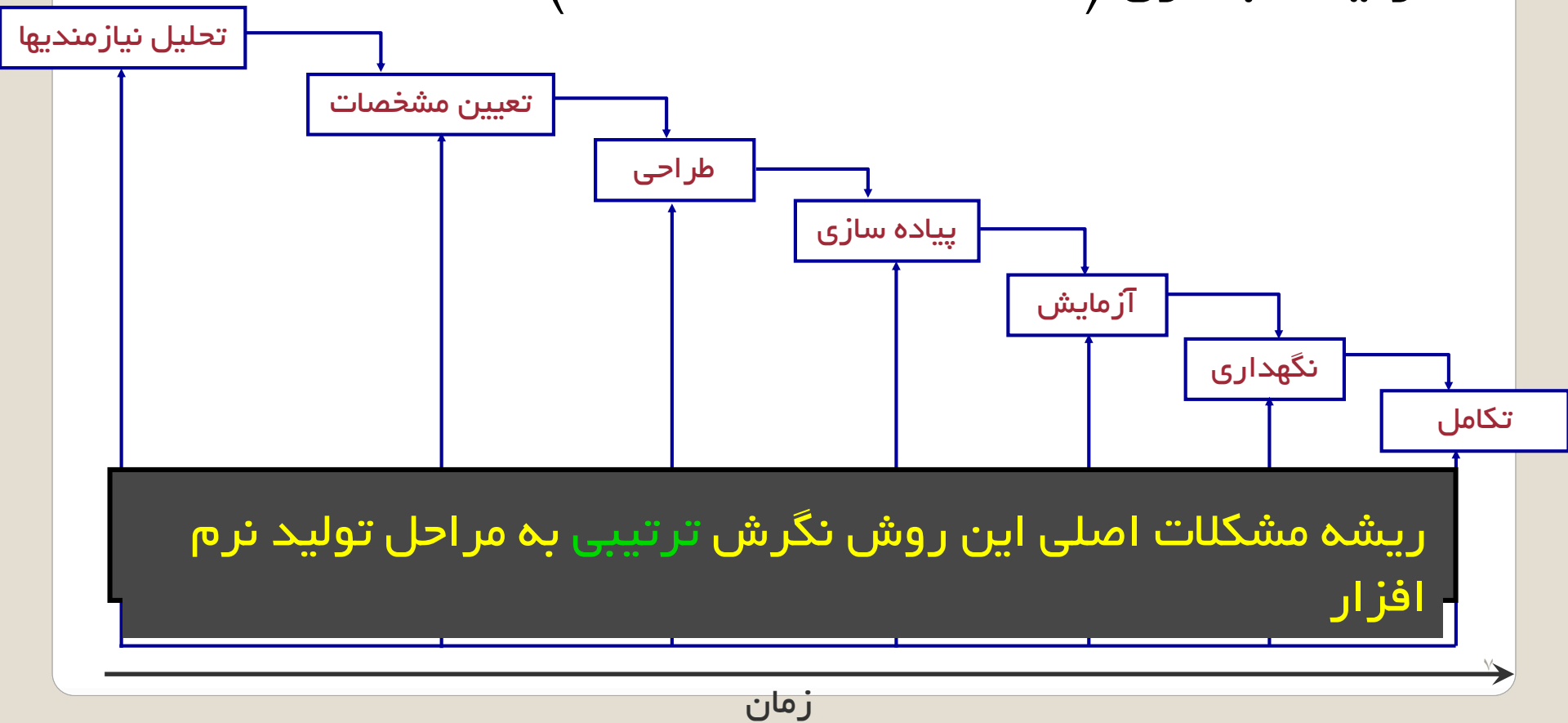
مدلسازی
تصویری
نرم افزار

بررسی
کیفیت
نرم افزار

کنترل تغییرات نرم افزار

تکرار و توسعه تدریجی

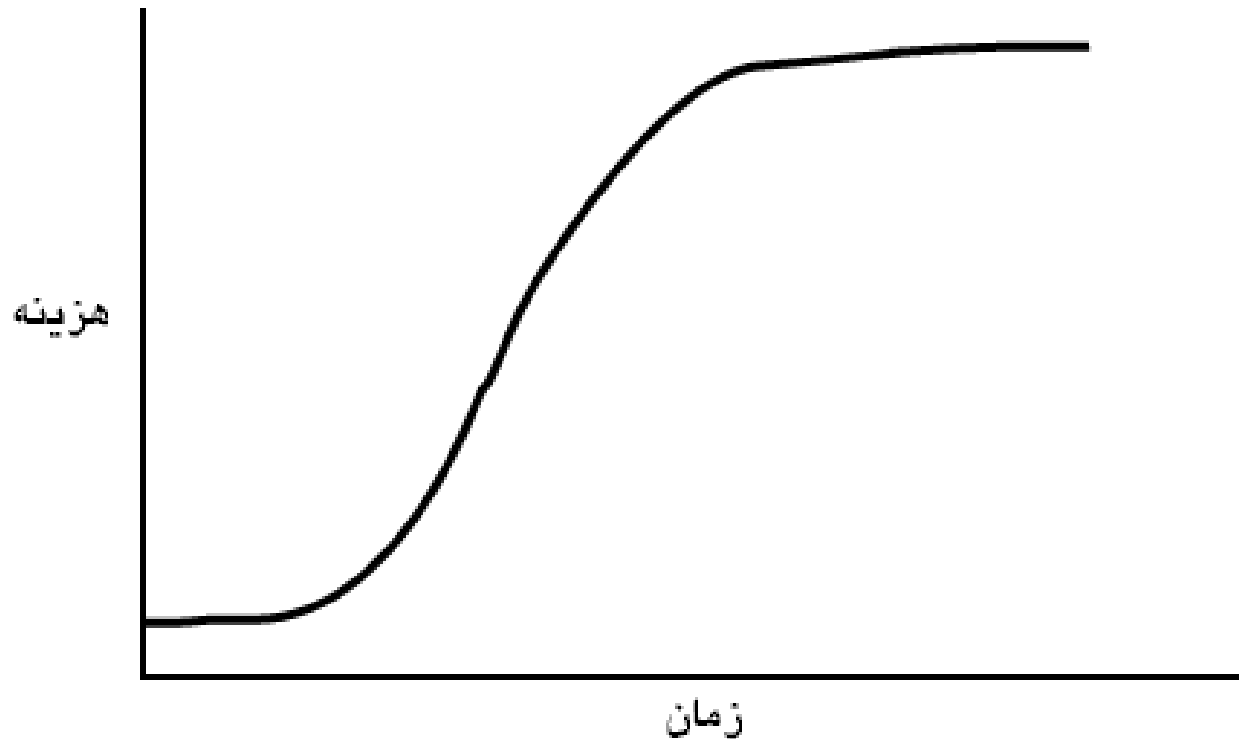
فرآیند آبشاری (Waterfall Process)



تکرار و توسعه تدریجی

(ادامه)

هزینه ریسک در روش آبخاری



تکرار و توسعه تدریجی

(ادامه)

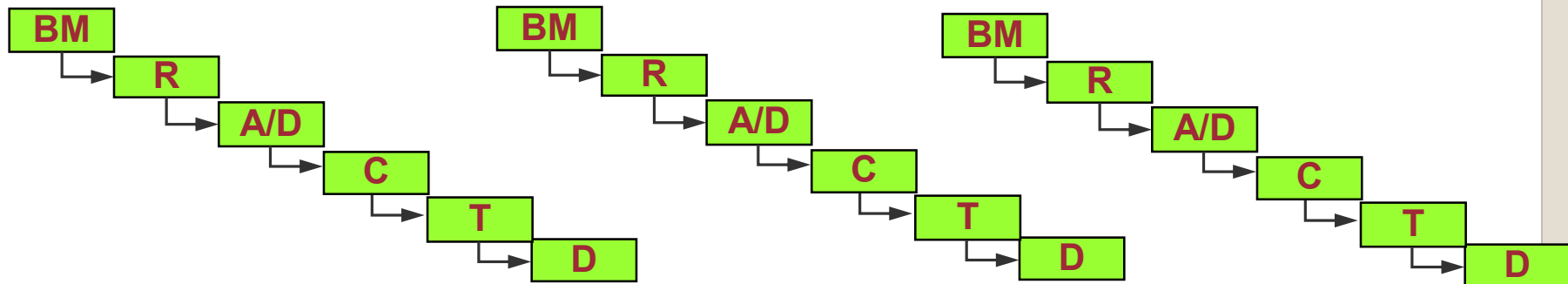
روش تکرار و توسعه
تدریجی

BM: Business Modeling
R: Requirements Analysis
A/D: Analysis & Design
C: Implementation
T: Test
D: Deployment

Iteration 1

Iteration 2

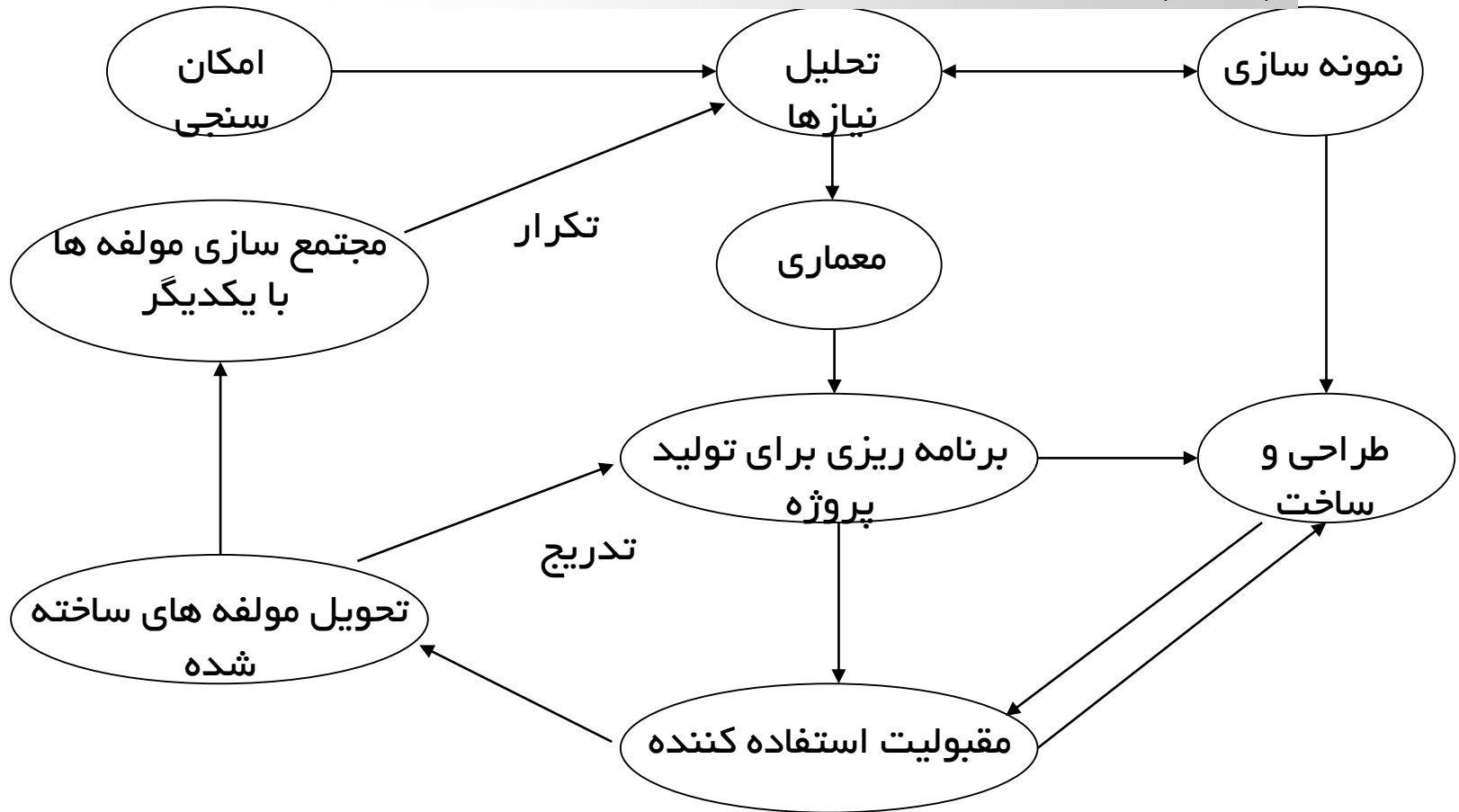
Iteration 3



T I M E

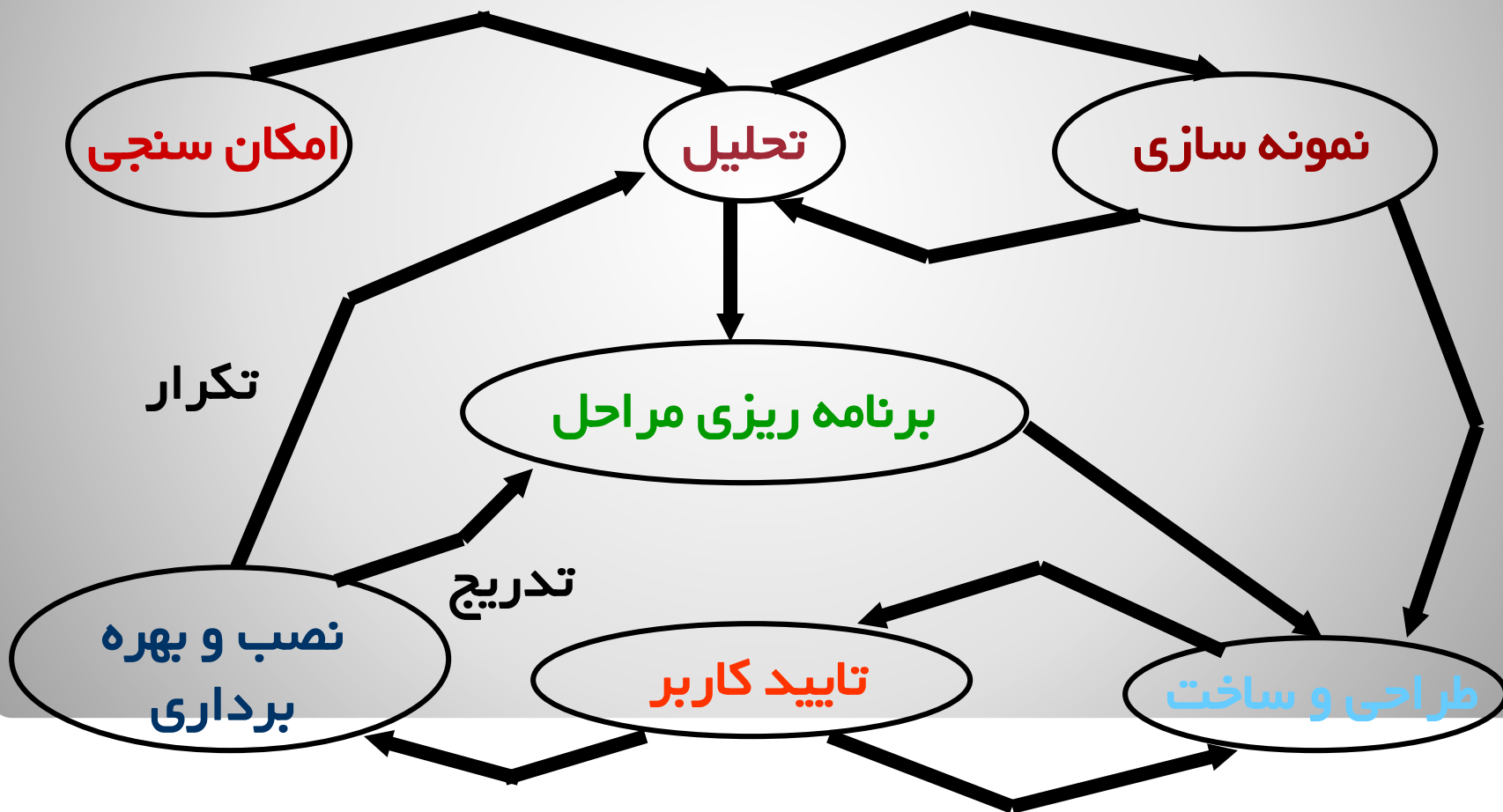
تکرار و توسعه تدریجی

(ادامه)



فرآیند تولید نرم افزار از دیدگاه شیء گرائی

فرآیند توسعه تکراری / تدریجی تولید نرم افزار به روش شیء گرا



تکرار و توسعه تدریجی

(ادامه)

ویژگیها:

- ❖ تشخیص زود هنگام خطاهایی که در درک مسأله، تحلیل یا طراحی رخ می دهند
- ❖ تشخیص زود هنگام ناسازگاری های موجود بین تحلیل نیازمندیها، طراحی و پیاده سازی
- ❖ کاربر می تواند دائما از روند پیشرفت پروژه مطلع گردد

تکرار و توسعه تدریجی

(ادامه)

ویژگیها: (ادامه)

❖ بوسیله این روش می توان روی قسمتهای مهمتر پروژه متمرکز شد و از پرداختن به قسمتهای کم اهمیتتر پرهیز نمود

❖ آزمایش تکراری و مستمر امکان تشخیص بهتر روند پیشرفت پروژه را به ما می دهد

❖ بارکاری (**Workload**) تیم ها، بخصوص آزمایش کنندگان، روی چرخه تولید پروژه به صورت متوازن توزیع می شود

مدیریت نیاز مندیها

”نیاز مندیها عبارتست از شرطی یا قابلیت که سیستم باید دارای آن باشد“

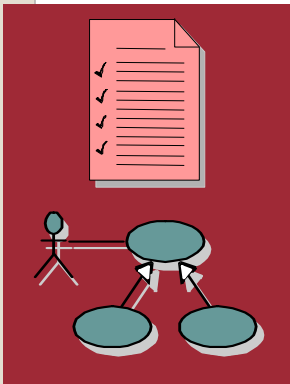
مدیریت نیاز مندیها شامل:

۱) دریافتن، سازمان دهی و مستند سازی عملکرد مطلوب سیستم و محدودیتهای موجود

۲) اعمال تغییرات مطلوب روی نیاز مندیهای جمع آوری شده

۳) ردیابی و مستند کردن اثرات بوجود آمده و تصمیم گیریهای اتخاذ شده

یکی از ثابتها در دنیای نرم افزار متغیر و پویا بودن نیاز مندیهاست



مدیریت نیاز مندیها

(ادامه)

ویژگیها:

- ❖ یک روش منظم و سیستماتیک برای مدیریت نیاز مندیها
- ❖ در این روش می توان نیاز مندیها را اولویت بندی، فیلتر بندی یا ردیابی نمود
- ❖ امکان تشخیص و افعی و منصفانه عملکرد و کارایی سیستم وجود دارد
- ❖ ناسازگاریها به آسانی قابل کشفند

استفاده از معماری مبتنی بر مؤلفه‌ها

”معماری سیستم عبارتست از تعیین ساختار کلی سیستم و روشهایی که این ساختار را قادر به تامین کلیه ویژگیهای کلیدی سیستم می سازد“

استفاده از معماری... (ادامه)

معماری سیستم شامل تصمیم گیری‌هایی در سطح کلان در موارد ذیل است:

- ❖ نحوه سازمان دهی سیستم نرم افزاری
- ❖ انتخاب عناصر ساختاری و واسط‌های آنها + مشخص نمودن رفتار این عناصر
- ❖ سازمان دهی این عناصر در گروه‌های بزرگتر (زیر سیستمها)
- ❖ سبک معماری مورد استفاده

به علاوه ساختار و رفتار سیستم، معماری با مواردی از قبیل کارایی، انعطاف پذیری، استفاده مجدد و محدودیت‌های تکنولوژی و اقتصادی نیز سروکار دارد

استفاده از معماری... (ادامه)

” یکی از شیوه های مهم معماری نرم افزار، توسعه مبتنی بر مؤلفه ها است زیرا این روش امکان استفاده مجدد از

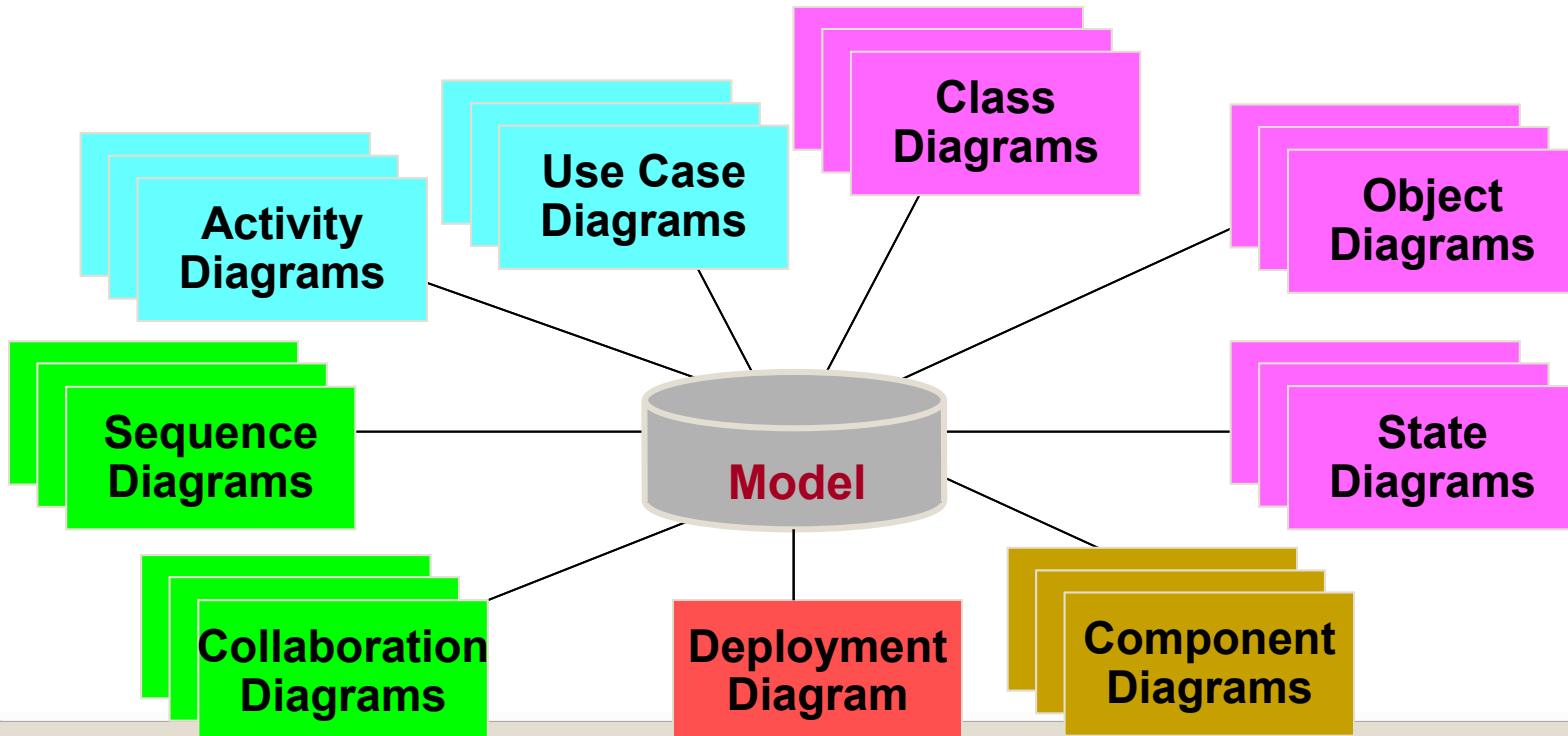
ویژگیها: مؤلفه های موجود را به ما می دهد“

- ❖ کمک در داشتن یک معماری کشسان
- ❖ افزایش قابلیت استفاده مجدد از مؤلفه ها و تکنولوژی های موجود
- ❖ مؤلفه یک پایه خوبی برای مدیریت پیکربندی است

مدلسازی تصویری

نرم افزار

”مدل عبارت از یک توصیف ساده شده، با توجه به یک نگرش معین، از سیستم است“



مدلسازی تصویری نرم افزار

ویژگی (ادامه)

❖ بوسیله مدل‌ها می‌توان طراحی سیستم را به صورت روشن بیان کرد

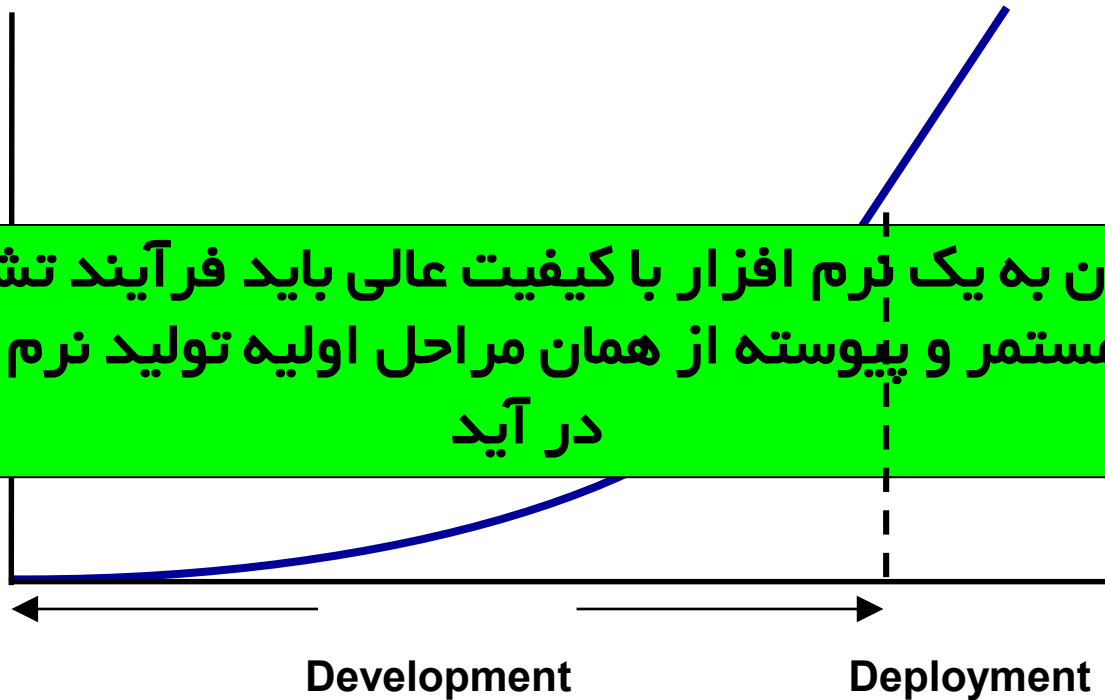
❖ امکان تشخیص معماریهای غیر قابل انعطاف و واحدبندی نشده بوسیله مدل‌ها

❖ امکان توصیف سیستم با میزان دلخواهی از جزئیات

❖ نرم افزار خوب حاصل مدل‌های با کیفیت بالا

بررسی کیفیت نرم افزار

برای رسیدن به یک نرم افزار با کیفیت عالی باید فرآیند تشخیص کیفیت به صورت مستمر و پیوسته از همان مراحل اولیه تولید نرم افزار به اجرا در آید



هزینه اصلاح خطاها به صورت نمایی رشد می نماید

بررسی کیفیت نرم افزار

ویژگی (ادامه)

- ❖ فرآیند تشخیص روند پیشرفت پروژه مبتنی بر واقعیت ها، نه بر حدسها و محاسبات کاغذی خواهد بود
- ❖ این فرآیند ناسازگاریهای موجود بین نیازمندیها، طراحی و پیاده سازی را آشکار می سازد
- ❖ امکان کشف زود هنگام خطاها را به ما می دهد و بدین صورت هزینه اصلاح آنها بشدت کاهش می یابد

مدیریت پیکربندی

”مدیریت پیکربندی عبارتست از هنر تشخیص،
سازماندهی و کنترل تغییراتی که برای نرم افزار در
مدت کارکرد خود (از ابتدای تولید تا خارج شدن از
عمل) رخ می دهند“

یکی از مشکلات اساسی تولید نرم افزار مدیریت پیکربندی
است. این مشکل بویژه در پروژه های بزرگ که در آن تیم های
متعددی با یکدیگر بر روی تکرارها، نشرها، محصولات و
سکوهای متفاوت کار می کنند، قابل مشاهده است

مدیریت پیکربندی (ادامه)

ویژگیها:

- ❖ یک روش سیستماتیک و قابل تکرار برای کنترل تغییرات نرم افزار
- ❖ کنترل انتشار تغییرات
- ❖ کاهش تداخل بین کار توسعه دهندگان که به صورت موازی با هم کار می کنند
- ❖ نرخ تغییر معیار مناسبی برای تشخیص وضعیت فعلی پروژه است

معرفی محورهای اصلی USDP

” USDP یک فرآیند تولید مهندسی نرم افزار است که یک روش سیستماتیک و منظم برای ترتیب انجام فعالیتها در یک پروژه نرم‌افزاری را پیشنهاد می‌نماید“

USDP از مدل شئ‌گرایی حمایت نموده و از روش‌های مدرن توسعه نرم‌افزار پشتیبانی می‌نماید

USDP دارای یک چارچوب فرآیند (Process Framework) است. بدین معنی که USDP، همه عناصر لازم برای تولید محصولات نرم‌افزاری از سیستم‌های سنتی و معمولی گرفته تا سیستم‌های هوشمند و سیستم‌های اطلاعاتی بزرگ را دربردارد

معرفی محورهای اصلی USDP (ادامه)

*Unified Software Development
Process(USDP)*

*Is Customized to
(By Rational)*

*Rational Unified
Process(RUP)*

*Is Customized to
(By My Enterprise)*

My Enterprise Process

معرفی محورهای اصلی USDP (ادامه)

سه محور اصلی USDP عبارتند از:

۱) راهبری بر مبنای موارد کاربری **Use Case Driven**

۲) محوریت قرار دادن معماری **Architecture Centric**

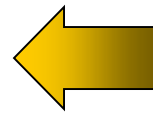
۳) استفاده از روش تکرار و توسعه تدریجی

Iterative and Incremental Development

راهبری بر مبنای موارد کاربری

”مورد کاربری بر مبنای موارد کاربری است که یک سیستم انجام می دهد تا یک نتیجه قابل مشاهده و ارزشمند برای کاربر فراهم نماید“

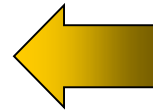
سیستم باید چه عملکردهایی از خود را نشان دهد؟



در نگرش

سستی

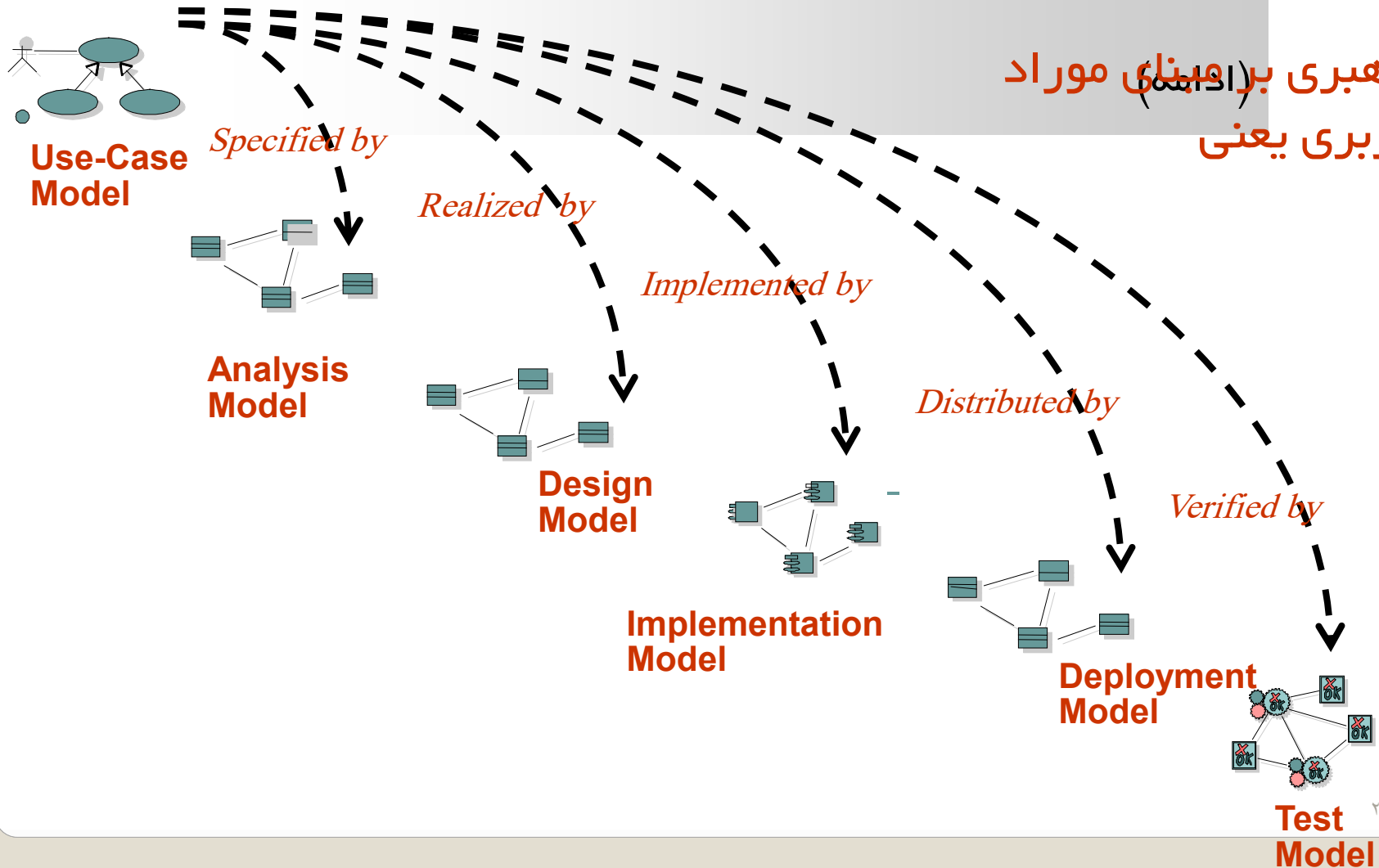
در نگرش مبتنی بر موارد کاربری



سیستم ، به ازای هر کاربر، باید چه عملکردهایی از خود را نشان دهد؟

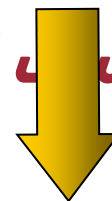
راهبری بر مبنای موارد کاربری

راهبری بر مبنای موارد
(ادامه)
کاربری یعنی



RUP بعنوان یک محصول

”چون مستندسازی RUP بوسیله اسناد کاغذی میسر
نبود، از اسناد HTML استفاده شده است“



- راهنماهای لازم برای بکارگیری RUP به عنوان یک فرآیند تولید در مراحل مختلف تولید نرم افزار
- راهنماهای ابزار Tools Mentors
- الگوها Templates
- یک Development Kit که چگونگی تغییر، گسترش، و تنظیم ویژه RUP را نشان می دهد

RUP بعنوان یک محصول (ادامه)

Requirements: Overview

Introduction | Workflows | Activities | Artifacts | Guidelines | Concepts

Overview
Phases
Core Workflows
Business Modeling
Requirements
Analysis & Design
Implementation
Test
Deployment
Configuration & Change
Project Management
Environment
Workers and Activities
Artifacts
Business Modeling Set
Requirements Set
Analysis & Design Set
Implementation Set
Test Set
Deployment Set
Configuration & Change
Project Management Set
Environment Set
Report Overview
Guidelines Overview
Examples Overview
Stereotypes Reference
Tool Mentors
Templates
White Papers
Work Guidelines
Web Resource Center
About the Unified Process

Copyright © 1987 - 2000 Rational Software Corporation

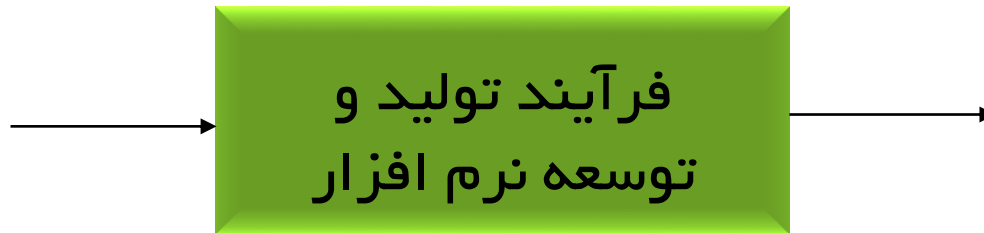
Start | Rational Unifi... | Presentation | Microsoft PowerP... | ch5.doc - Micro... | RationalUnifiedPr... | En 12:44

معرفی ابعاد فرآیند RUP

می توان نگاه‌های ذیل را به فرآیند داشت:

(۱) **جعبه سیاه** **Black Box**

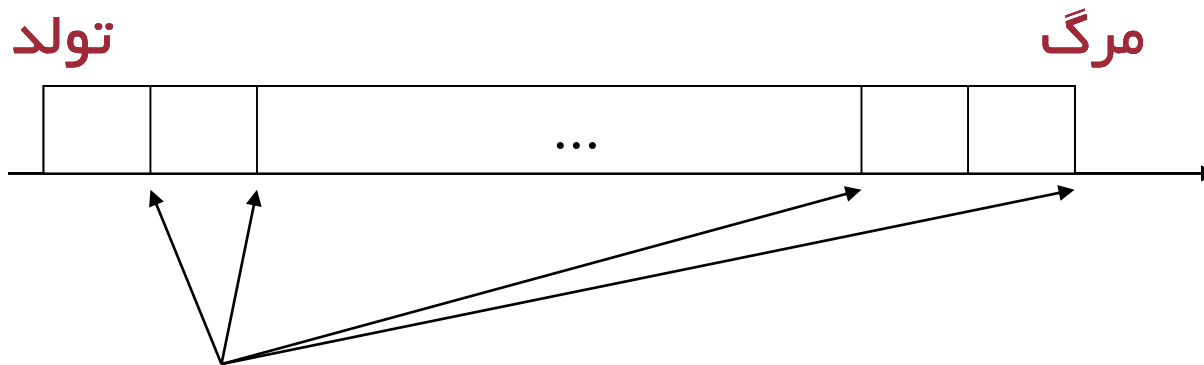
نیازمندیهای
جدید یا تغییر
یافته



محصول
جدید یا
تغییر یافته

معرفی ابعاد فرآیند RUP (ادامه)

(۲) با توجه به مدت کارکرد مفید سیستم نرم‌افزاری

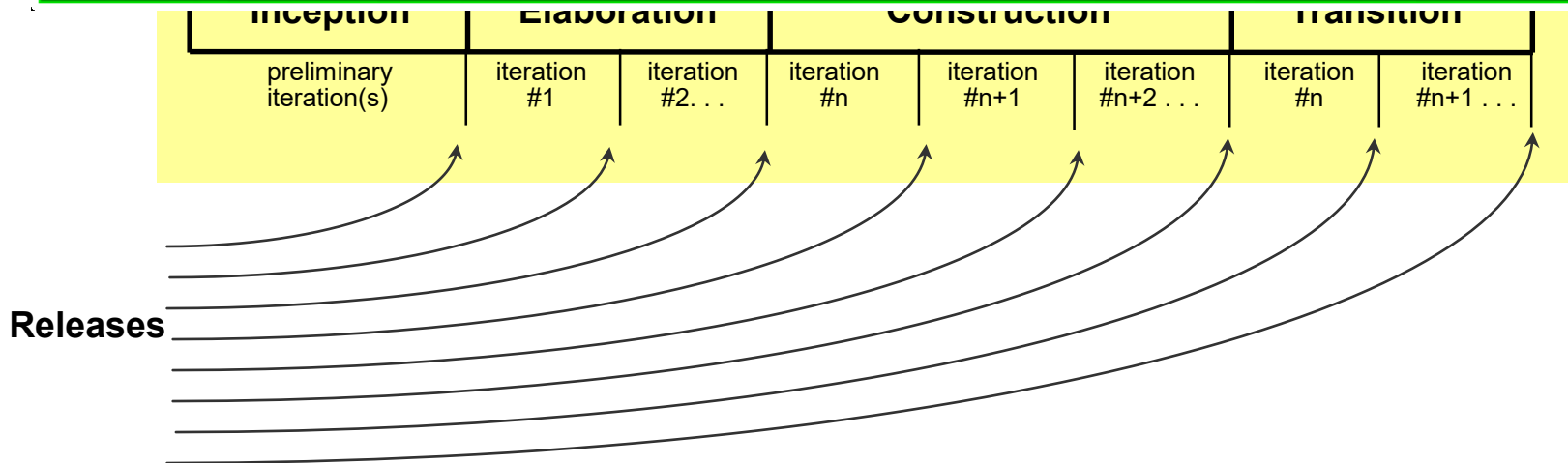


نسخه‌های یا نشرها (Releases)
مختلف

معرفی ابعاد فرآیند RUP (ادامه)

” در *RUP* دوران حیات یک نرم افزار به چهار مرحله آغازین، تشریح، ساخت، و انتقال تقسیم می شود“

سه مرحله اول شامل فعالیتهای تولید یا توسعه نرم افزار بوده و مرحله چهارم دربردارنده انتقال نرم افزار به محیط واقعی و نگهداری آن است



معرفی ابعاد فرآیند RUP (ادامه)

۳ دیدگاه دو بعدی

نرم افزار یک محصول فیزیکی که یک بار تولید و مستهلک شود
نیست،
بلکه

مانند یک موجود زنده‌ای است که در سازمان تولد و رشد کرده
و در دوران حیات خود باید با تغییر نیازهای سازمان و اهداف و
ماموریت‌های

آن خود را تطبیق دهد

معرفی ابعاد فرآیند RUP (ادامه)

مثال: زندگی انسان!

• تولد و مرگ دارد

می‌توان دوران حیات نرم‌افزار را به زندگی انسان تشبیه کرد!!!

دوران حیات یک سیستم نرم‌افزاری شامل دوره‌های گوناگونی است که میان آنها فعالیت‌های مشترکی وجود داشته که در هر دوره به اندازه معینی روی هر کدام از این فعالیتها تاکید می‌شود. همچنین هر کدام از این دوره‌ها می‌توانند فعالیت‌های ویژه خود را داشته باشد.

معرفی ابعاد فرآیند RUP (ادامه)

*” RUP یک فرآیند تولید دو بعدی است(بر عکس
فرآیندهای تولید سنتی که یک بعدی هستند)“*

این ابعاد عبارتند از:

**۱) بعد (محور) عمودی: این محور گردش کارهای اصلی
را نشان می دهد**

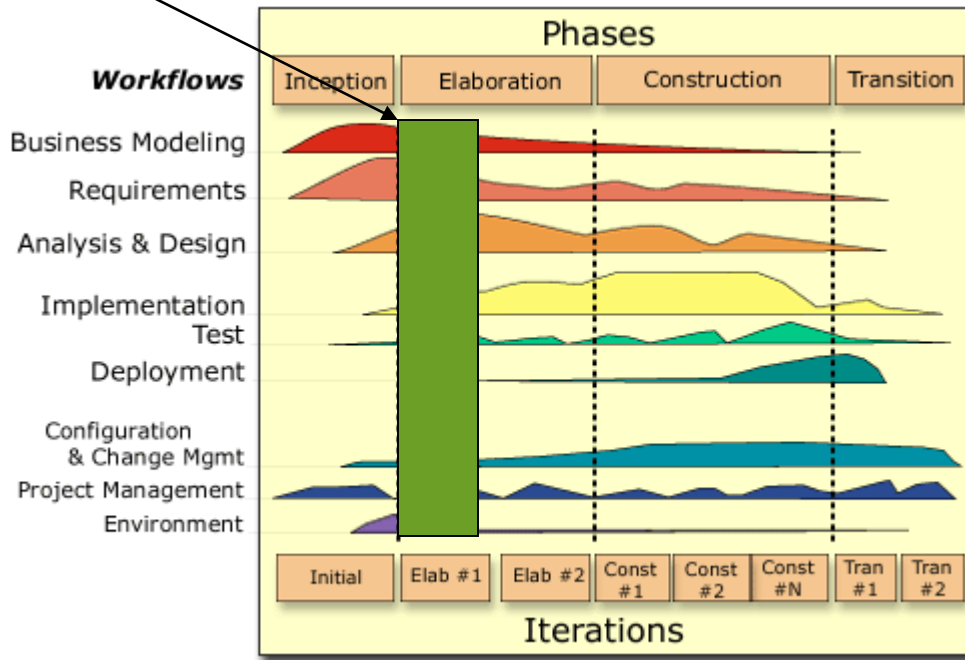
**۲) بعد (محور) افقی: این محور ساختار چرخه تولید نرم
افزار در RUP در بستر زمان را نشان می دهد**

معرفی ابعاد فرآیند RUP (ادامه)

فرآیند دو بعدی

در یک تکرار همه
گردش کارها اجرا
می شوند

فعالیت‌های
مشترک



زمان

Component-Based Development

