



Agile Development

درس مهندسی نرم افزار ۲



Agile manifesto

- اهمیت اشخاص و ارتباطات نسبت به فرایند و ابزار: اهمیت شخص تولید کننده و روش تولید و ارتباطات بین این اجزا نسبت به ابزار و فرآیند (بر عکس شی گرا)
- اهمیت نرم افزار عملیاتی و تولید کد نسبت به مستندات کامل (افزایش هزینه نگهداری سیستم)
- اهمیت همکاری مشتری و کاربر در تولید سیستم نسبت به مذاکرات در حین قرارداد (JAD)
- اهمیت پاسخ به تغییرات نسبت به پیگیری یک طرح اولیه: ارجحیت تغییرپذیری سیستم نسبت به برنامه تنظیم شده برای سیستم
 - موجب افزایش کارایی و کاهش استحکام برنامه تولید می شود.

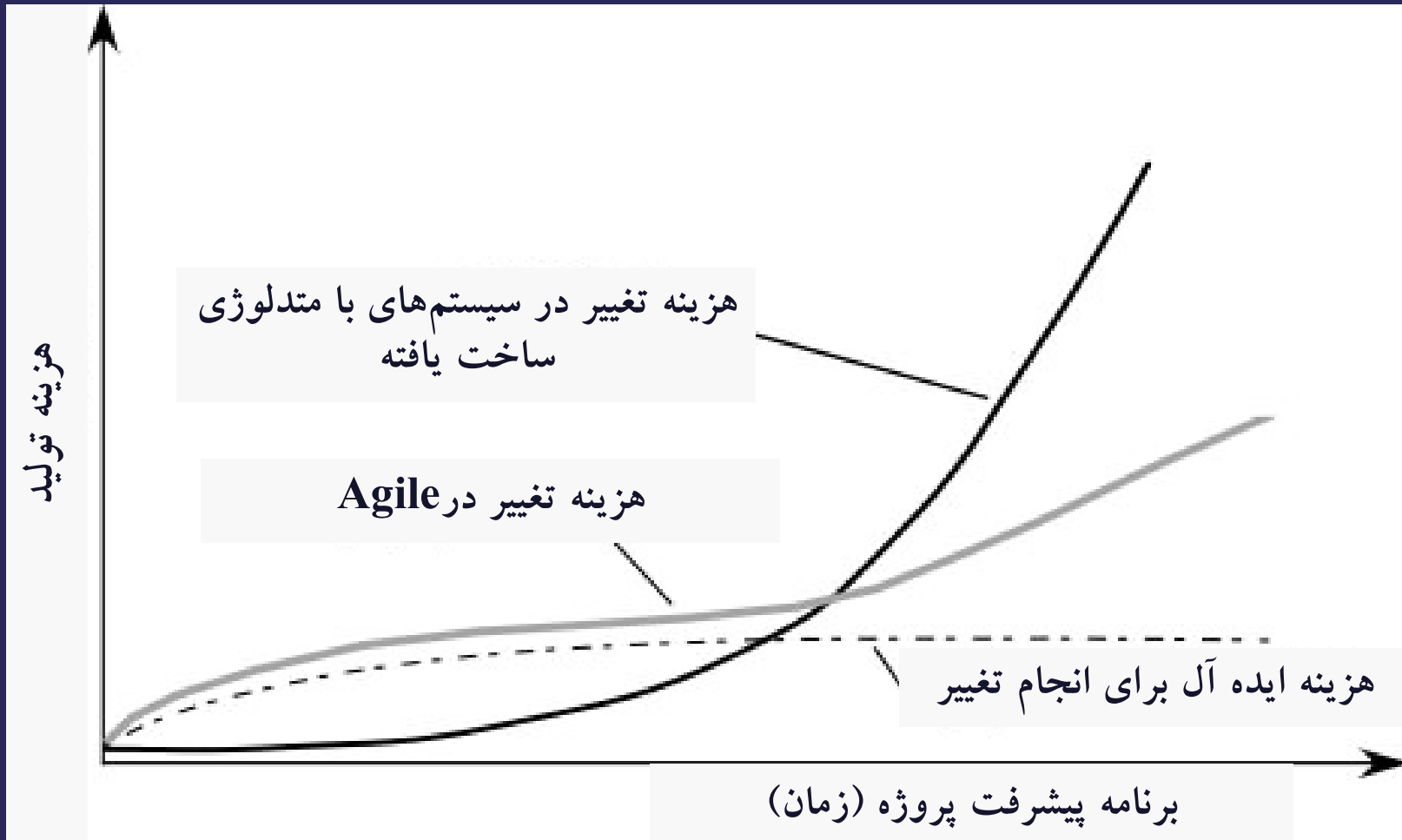


Agile

- در Agile تاکید بر تغییرپذیری و adaptive بودن مساله اصلی است.
 - Agile نیازمند ایجاد رابطه همیشگی و موثر بین تولید کننده و استفاده کننده نرم افزار است.
 - برای این منظور کاربر در تیم تولید وجود دارد و در رابطه با کیفیت و عملکرد نرم افزار دارای نظر است.
- ← تحویل سریع و افزایشی (Incremental) نرم افزار



Agility and the Cost of Change





Agile methodologies

- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Dynamic Systems Development Method (DSDM)
- Scrum
- Crystal
- Feature Driven Development (FDD)



ASD

- توسط Jim Highsmith در سال ۲۰۰۲ مطرح شد.
- تمرکز آن بر همکاری افراد و خود سازمان دهی (self-organization) است.
- ویژگی های اصلی ASD:
 - Mission focused: در بسیاری از مواقع نیازمندی ها به طور کامل مشخص نیستند اما mission کلی مشخص است. mission مرزهای سیستم را تعیین و تصمیمات مورد اتخاذ را جهت دهی می کند.
 - Feature based
 - Iterative
 - Time-boxed planning: زمان مورد نیاز برای انجام یک task/activity مشخص شده است.
 - Risk driven: توجه به ریسک های پروژه مورد تاکید است.
 - Change tolerant
 - Component based: مورد تاکید است.



ASD

adaptive cycle planning
mission statement
project constraints
basic requirements
time-boxed release plan

Requirements gathering
JAD

Speculation

Collaboration

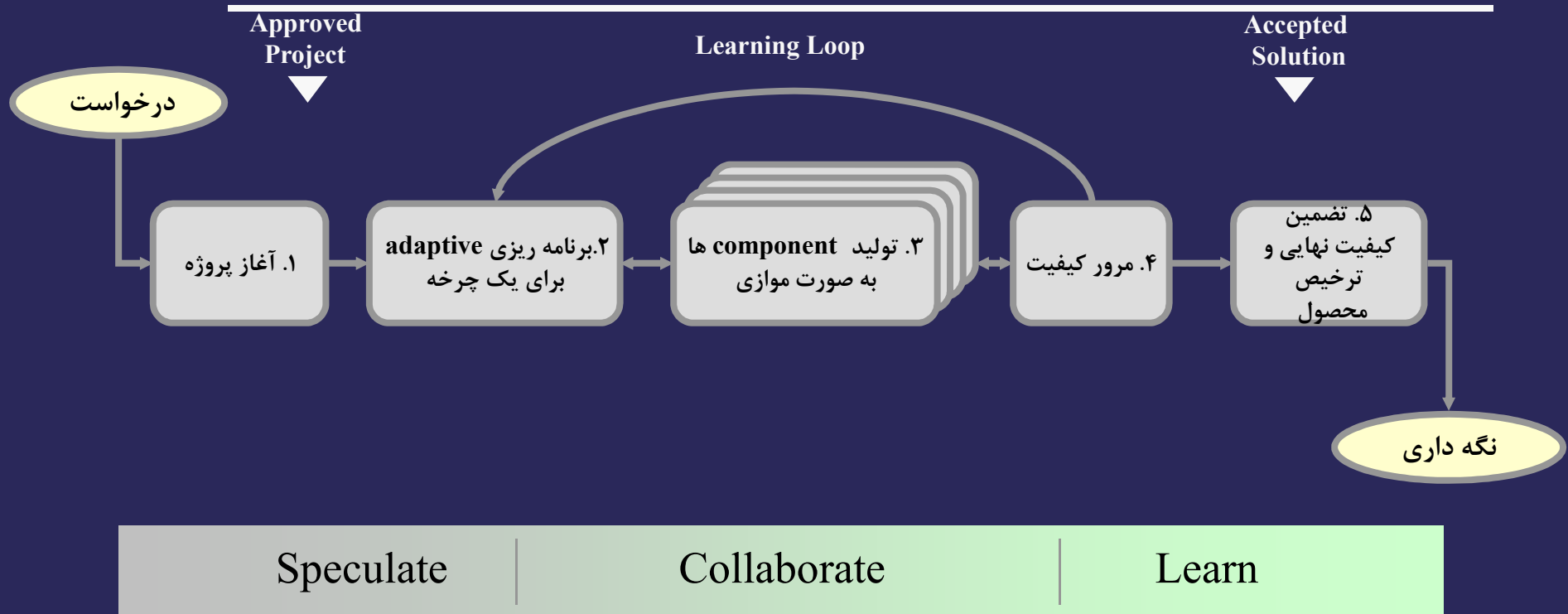
Release

Learning

components implemented/tested
focus groups for feedback
formal technical reviews
postmortems



ASD



Legend:



Milestone



Project
Process



Parallel
Processes



Preceding or
Succeeding
Process



ASD

➤ آغاز و برنامه ریزی پروژه:

- شناسایی اهداف و مأموریت پروژه
- شناسایی قیود، شناسایی و جمع آوری نیازمندی‌ها
- تخمین سائز و scope پروژه، شناسایی ریسک‌های کلیدی پروژه
- ✓ اطلاعات مربوط به آغاز پروژه از طریق جلسه JAD جمع‌آوری می‌شوند.



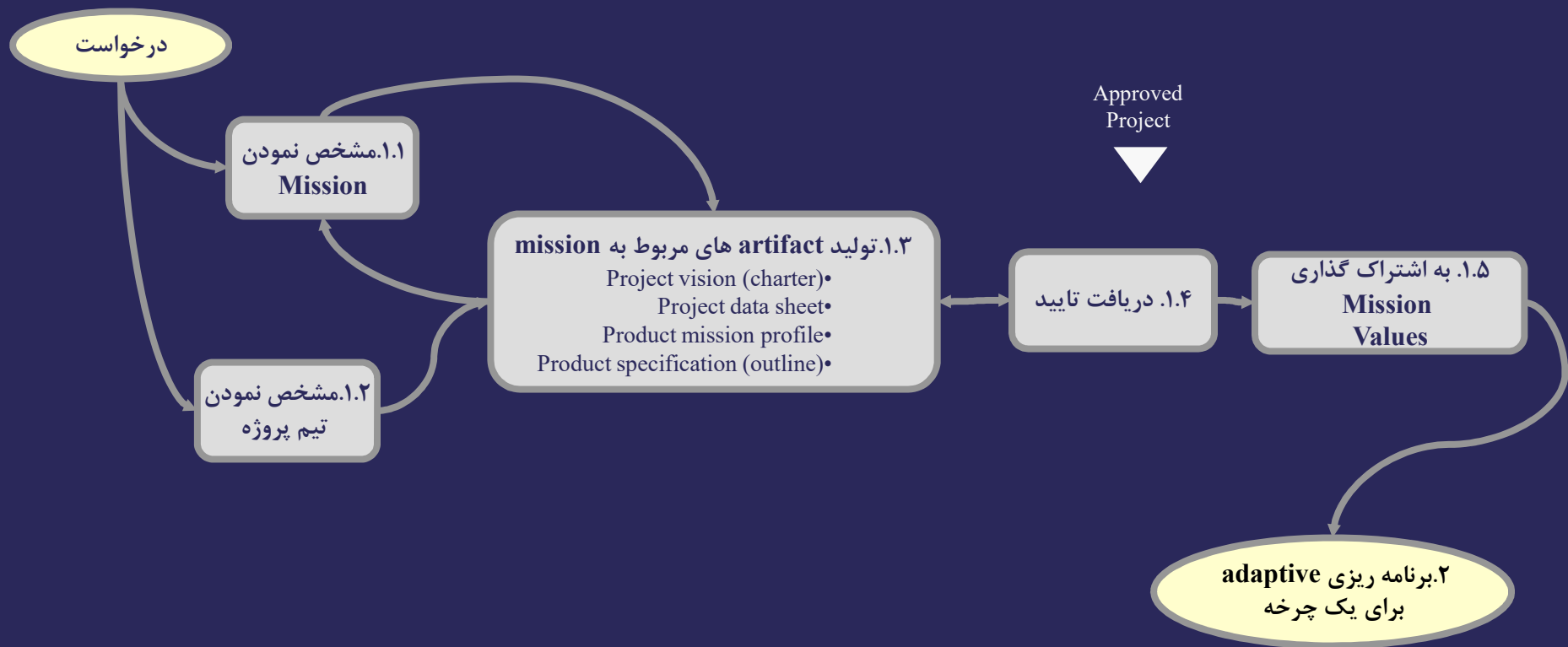
ASD

➤ برنامه ریزی adaptive برای یک چرخه

- مشخص نمودن time-box پروژه، تعداد iteration های مورد نیاز،
مشخص نمودن time-box برای iteration های پروژه
- مشخص نمودن component هایی که باید تولید شوند.
- اختصاص component ها به هر iteration

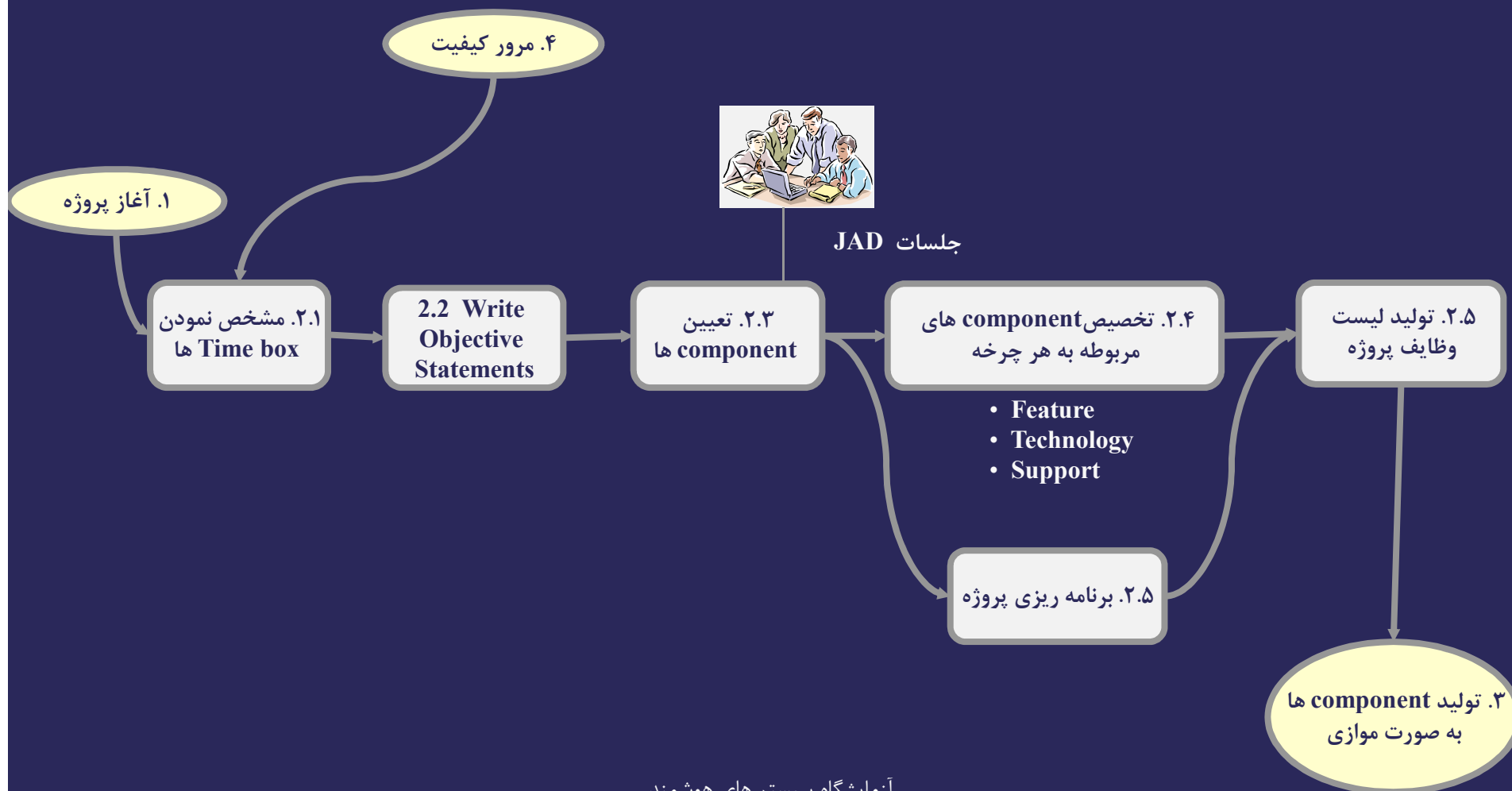


ASD



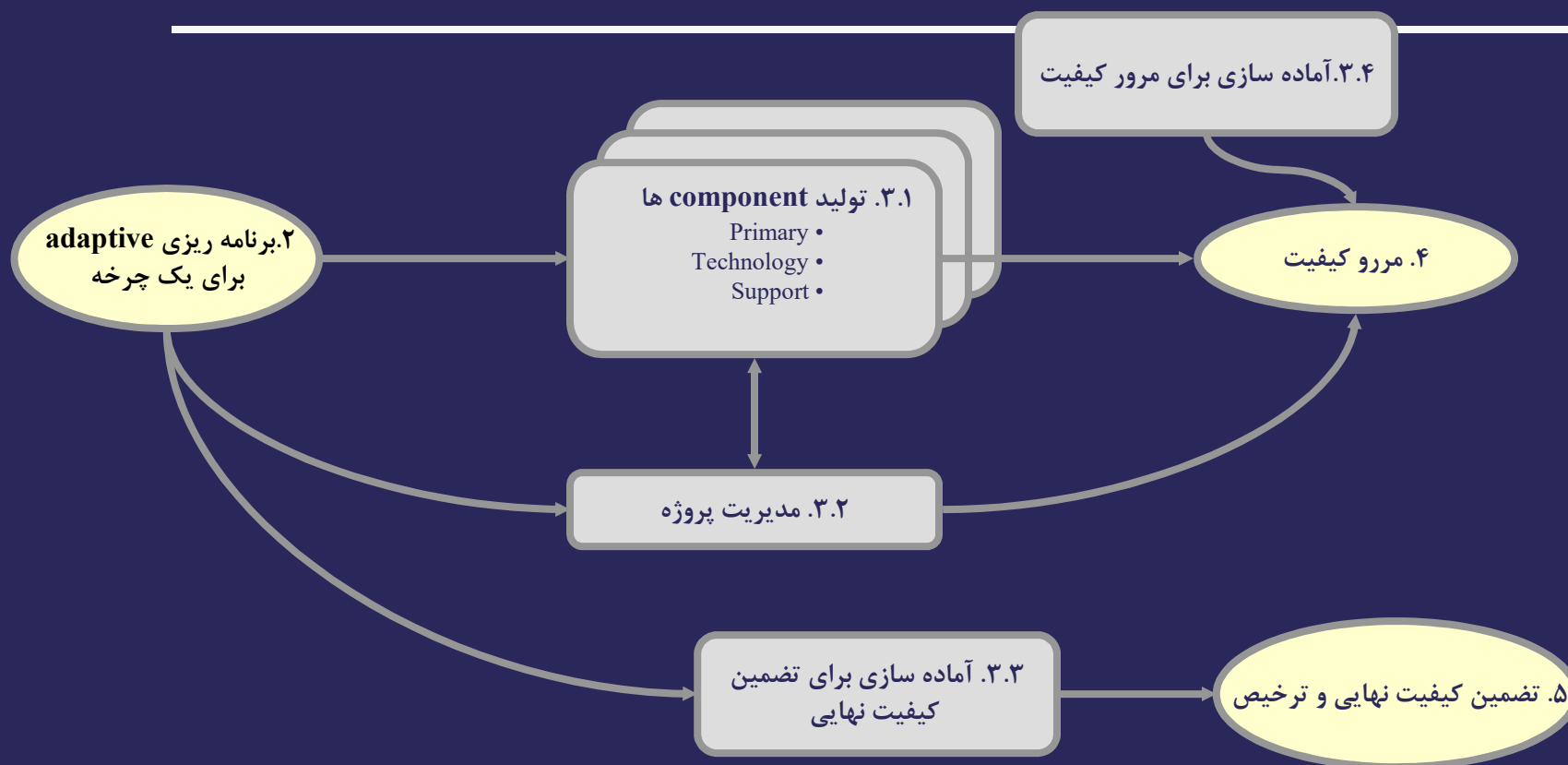


ASD



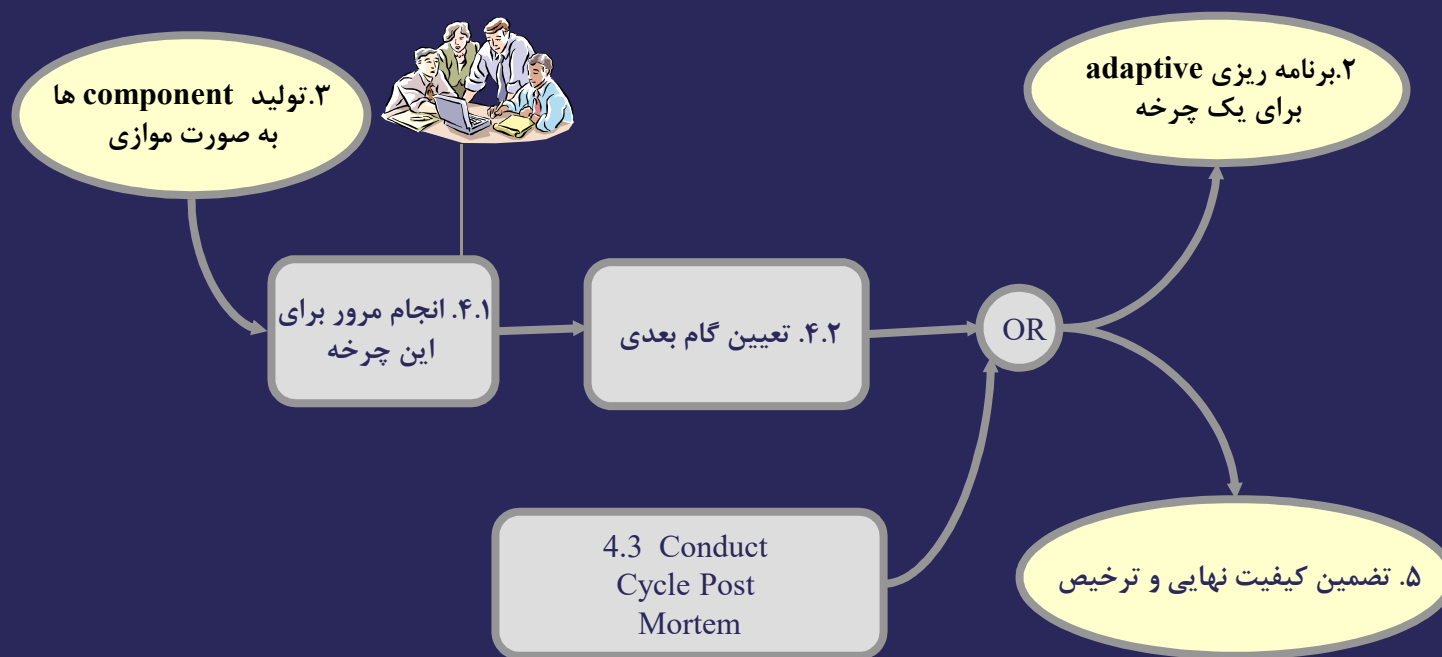


ASD



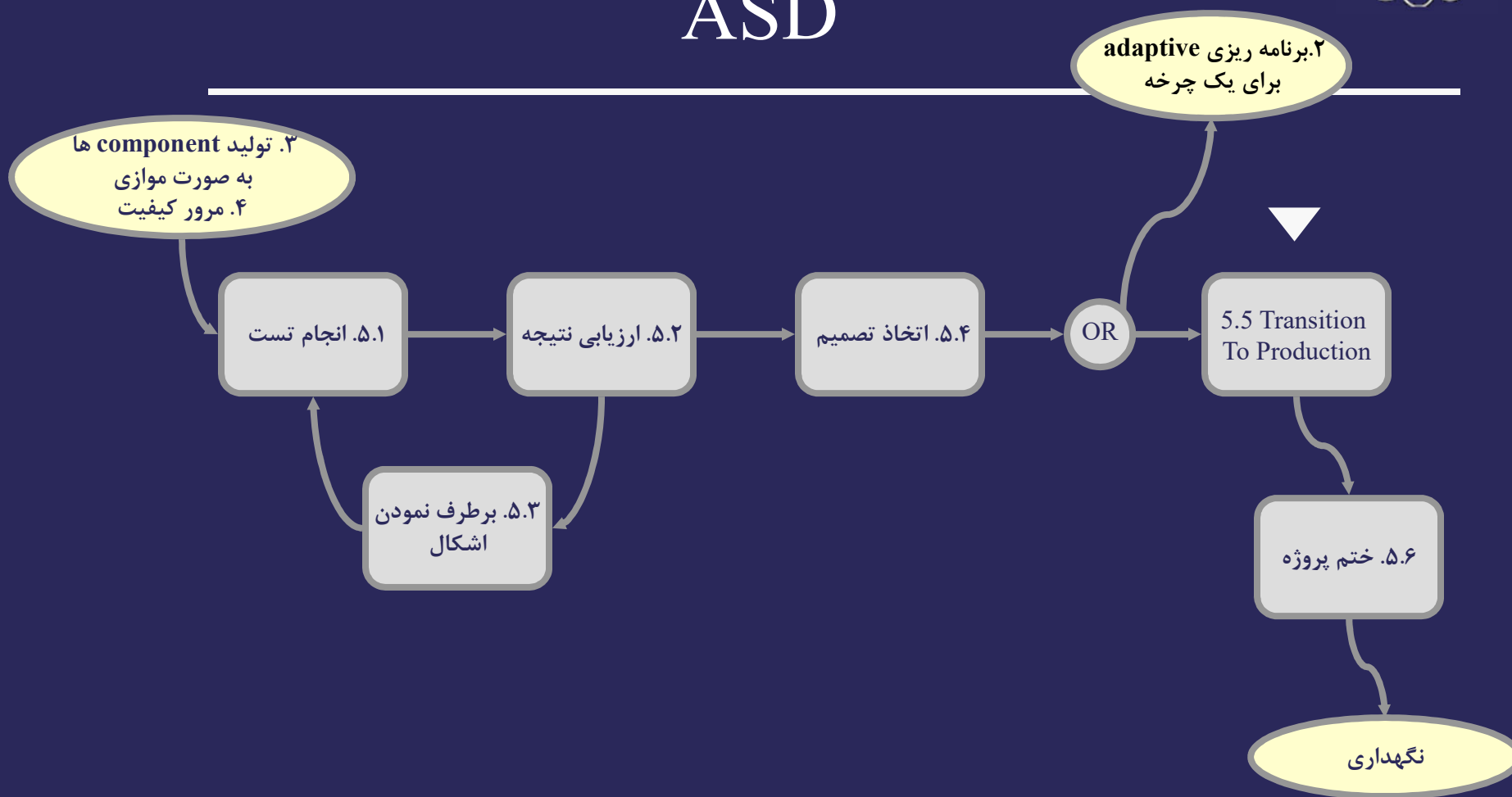


ASD





ASD



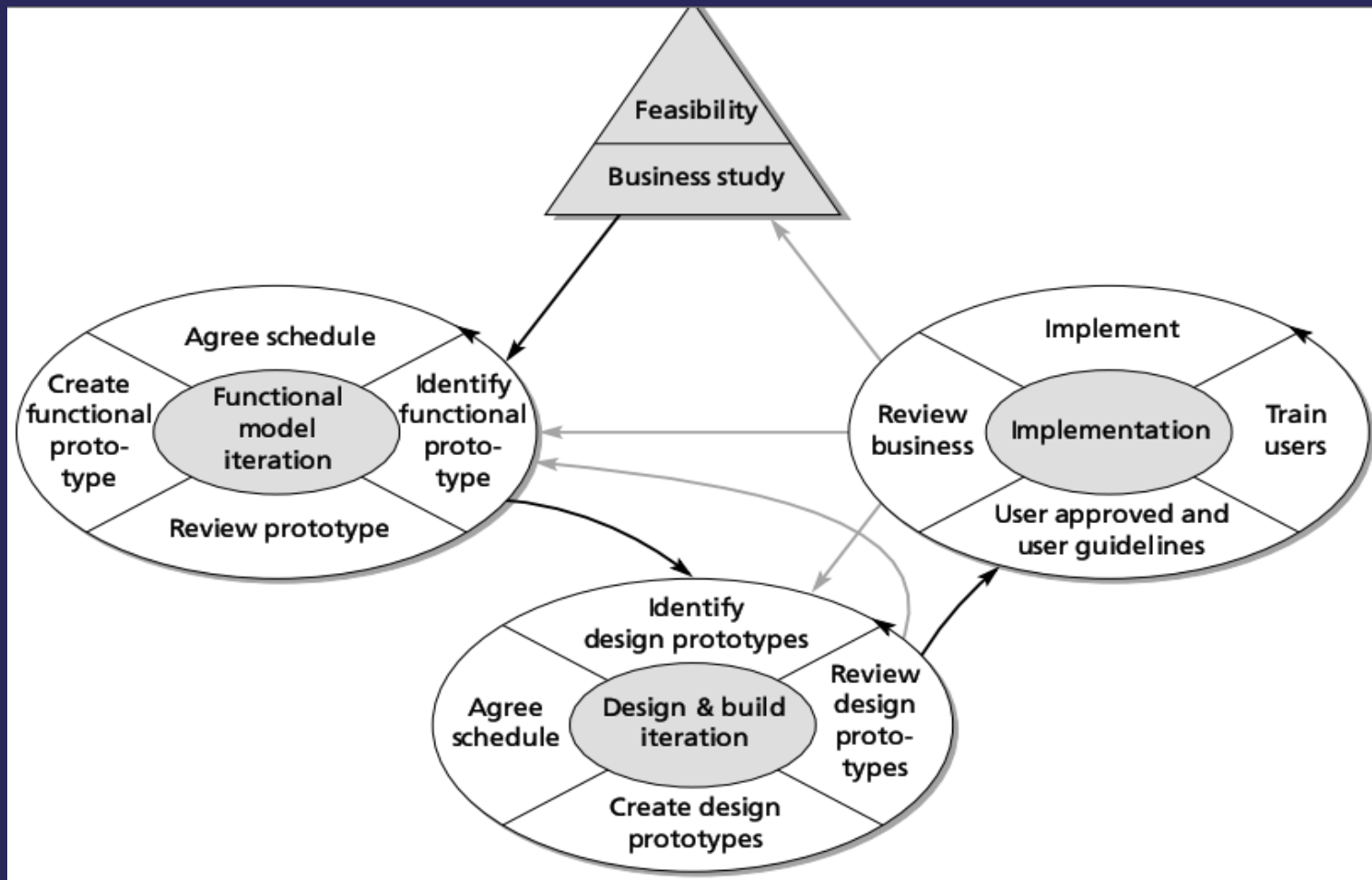


DSDM

- تاکید بر incremental prototyping
- به صورت iterative بوده و در هر iteration بر مبنای قانون pareto کار می‌کند. (۸۰٪ محصول در ۲۰٪ زمان مورد نیاز برای تولید آن قابل تحویل است.)
- به این ترتیب در هر iteration، تنها به اندازه‌ای کار می‌شود که بتوان به iteration بعدی رسید.
- سایر جزئیات را می‌توان زمانی که نیازمندی‌های بیشتری شناسایی شد و یا تغییراتی درخواست گردید، کامل نمود.



DSDM





DSDM

Feasibility study ➤

- مانند تحلیل امکان‌سنجی است که به صورت کلی در آغاز تمام پروژه‌ها انجام می‌پذیرد. با این تفاوت که تمرکز بیشتر بر بررسی مناسب بودن متدولوژی DSDM برای این پروژه است. (آیا DSDM بهترین متدولوژی برای این پروژه است؟)

Business study ➤

- آشنایی با کسب و کار، نیازمندی‌های کارکردی و اطلاعاتی
- تاکید بر collaborative بودن این مرحله (برگزاری workshop)
- تعریف معماری اولیه
- ارائه برنامه تولید



DSDM

Functional model Iteration ➤

- انتخاب نیازمندی‌ها بر مبنای اولویت آن‌ها
- تحلیل و مدل‌سازی نیازمندی‌های منتخب بر مبنای پروتوتایپ‌های تکاملی
- جمع‌آوری سایر نیازمندی‌ها و دریافت feedback از طریق پروتوتایپ‌ها

Design & Build Operation ➤

- تکامل پروتوتایپ‌ها تا زمانی که قابل استقرار در محیط عملیاتی شوند
- انجام تست
- امکان موازی سازی آن با مرحله‌ی قبل وجود دارد.

Implementation ➤

- استقرار نسخه‌های افزایشی در محیط عملیاتی
- نسخه تحویل داده شده ۱۰۰٪ کامل نیست، همچنین امکان درخواست تغییر برای آن وجود دارد. در هر یک از این دو حالت (تکمیل و یا تغییر) به مرحله‌ی Functional model Iteration باز می‌گردیم.



DSDM

- بین DSDM و ASD /XP شباهتهایی وجود دارد.
- اما اصول اجرایی در آنها متفاوت است.
- این امکان وجود دارد که DSDM با XP ترکیب شود
- برای مثال بهره گیری از pair programming، تولید تست پیش از آغاز کد زنی و ...

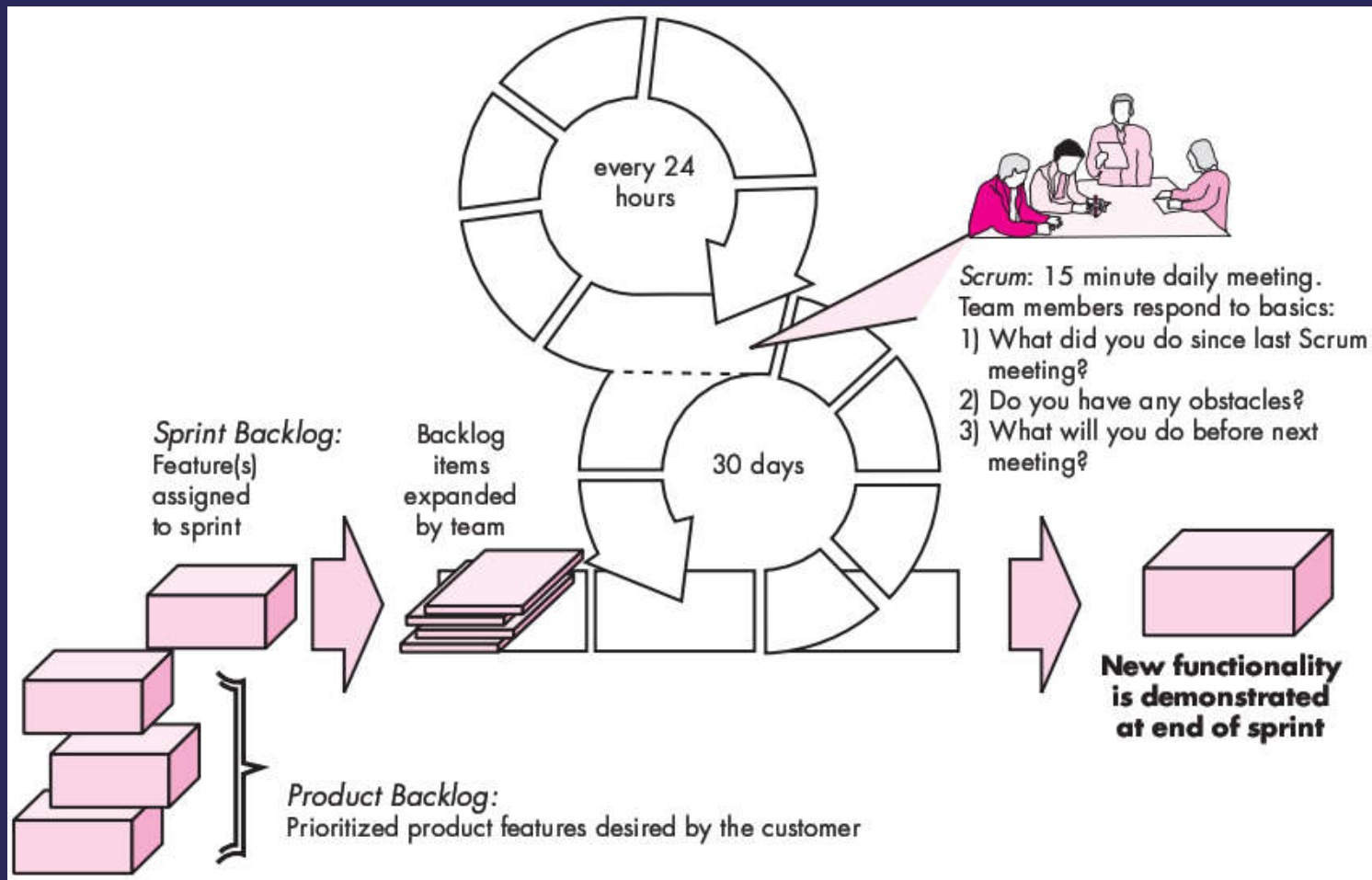


Scrum

- توسط Schwaber and Beedle ارائه شده است.
- Task های موجود در هر یک از framework activities بر مبنای یک سری process pattern انجام می پذیرند.
- این process pattern ها عبارتند از:
 - Backlog
 - Sprints
 - Scrum meetings
 - Demos
- کار تولید محصول به یک سری packet (sprint) تقسیم می گردد.
- تعداد sprint های مورد نیاز برای هر framework activity بسته به سائز و پیچیدگی پروژه متفاوت است.



Scrum





Scrum

Backlog ➤

- لیستی اولویت بندی شده از نیازمندی‌ها یا feature ها

Sprints ➤

- شامل واحدهای کاری مورد نیاز برای دستیابی به نیازمندی‌های تعریف شده در backlog است (time-box آن معمولاً ۳۰ روزه است)
 - تغییرات (مثلاً تغییر در backlog) در یک sprint تاثیرگذار نیستند، بنابراین اعضای تیم در بازه‌های کوتاه مدت اما در محیطی stable کار می‌کنند.
- Scrum meetings: جلسات کوتاه (معمولاً ۱۵ دقیقه ای) که روزانه توسط اعضای تیم برگزار می‌شود. (پیدا کردن مشکلات احتمالی در سریع ترین زمان ممکن)
- Demos: تحویل increment به مشتری



Crystal

➤ متشکل از خانواده‌ای از متدلوژی‌هاست و در اواسط دهه ۱۹۹۰ توسط Cockburn ارائه گردید.

➤ تمرکز آن بر روی موارد زیر است:

- People
- Interaction
- Community
- Skills
- Talents
- Communications

✓ تاکید بر ارتباطات face to face



Crystal Families

➤ این متدلوژی متشکل از خانواده‌های از متدلوژی‌هاست که عبارتند از: (براساس اندازه و حساسیت سیستم متدلوژی مناسب تعیین می‌شود).

Small projects , no mission critical



Large projects , no mission critical

- Crystal Clear ●
- Crystal Yellow ●
- Crystal Orange ●
- Crystal Orange Web ●
- Crystal Red ●
- Crystal Maroon ●
- Crystal Diamond ●
- Crystal Sapphire ●



Crystal family: Project criticality and size

	Clear	Yellow	Orange	Red	Maroon
Life (L)	L6	L20	L40	L80	L200
Essential Money (E)	E6	E20	E40	E80	E200
Discretionary Money (D)	D6	D20	D40	D80	D200
Comfort (C)	C6	C20	C40	C80	C200
	1-6	7-20	21-40	41-80	81-200

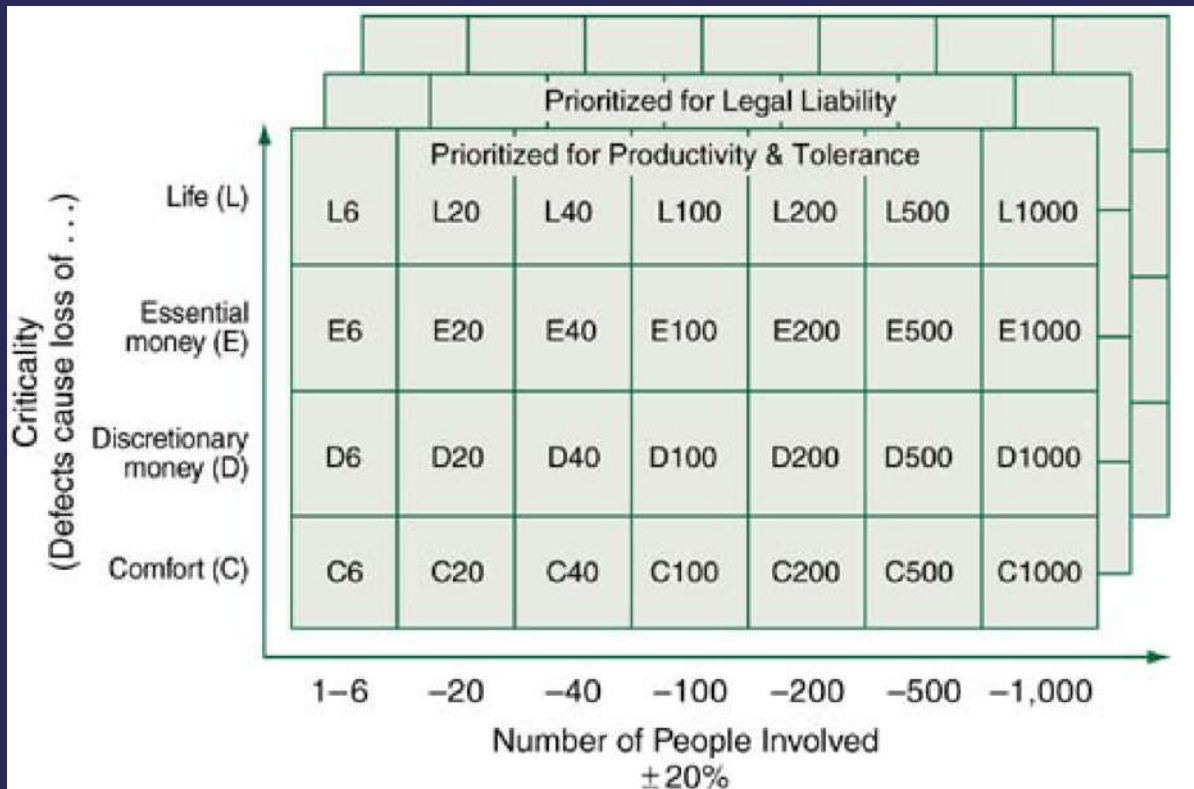
Project size

➤ در این متدلوژی ۴ سطح از حساس بودن تعیین می شود:

- Life ●
- Essential money ●
- Discretionary money ●
- Comfort ●



Crystal family: project priorities



time to market ➤

cost reduction ➤

exploration ➤

legal liability ➤



Crystal Families: common properties



➤ تحویل مکرر (Frequent delivery)

- مانند دیگر متدلوژی‌های agile، تحویل محصول به صورت iterative
- تحویل‌ها ممکن است به صورت هفتگی یا ماهیانه باشند بسته به نوع پروژه.
- Error ها سریع تر پیدا شده و برطرف می‌شوند.

➤ بهبود انعکاسی (Reflective improvement)

- توصیه به برگزاری Reflection workshop هر دو هفته یک بار جهت دریافت بازخورد توسط developer ها در رابطه با process ها (پیدا کردن process هایی که درست کار نمی‌کنند در راستای بهبود فرآیند تولید)

➤ ارتباطات نزدیک

- تاکید بر حضور تیم تولید در مکان یکسان
- در crystal clear و پروژه‌های کوچکتر، تیم‌های تولید در یک اتاق کار می‌کنند.
- کاهش سربار ارتباطی بین اعضای تیم تولید



Crystal Families: common properties



- امنیت پرسنل
- تمرکز
- دسترسی آسان به کاربران خبره
- تست‌های خودکار، مدیریت پیکربندی و یکپارچه سازی مداوم



FDD

- توسط Peter Coad در سال ۱۹۹۹ ارائه شد.
- تاکید بر روی feature
 - feature: یک کارکرد مورد نظر کاربر (client-valued function) است که امکان پیاده‌سازی آن در دو هفته وجود داشته باشد.
- فواید تولید بر مبنای feature
 - به دلیل اینکه feature ها بخش‌های کوچکی از کارکردهای قابل تحویل هستند، کاربر ساده‌تر آن‌ها را توصیف می‌نماید و مرور آن‌ها نیز آسان‌تر انجام می‌پذیرد.
 - feature ها می‌توانند در یک سلسله مراتب قرار گیرند و برنامه ریزی، زمان بندی و ردیابی پیشرفت پروژه بر مبنای این سلسله مراتب انجام پذیرد.



FDD:Feature template

➤ برای تعریف feature, template زیر پیشنهاد شده است:

● $\langle \text{action} \rangle$ the $\langle \text{result} \rangle$ $\langle \text{by for of to} \rangle$ a(n) $\langle \text{object} \rangle$

● مثال:

○ Add the product to shopping cart

○ Display the technical-specifications of the product

○ Store the shipping-information for the customer

➤ یک feature set, feature هایی را که از لحاظ کسب و کار مرتبطاند در

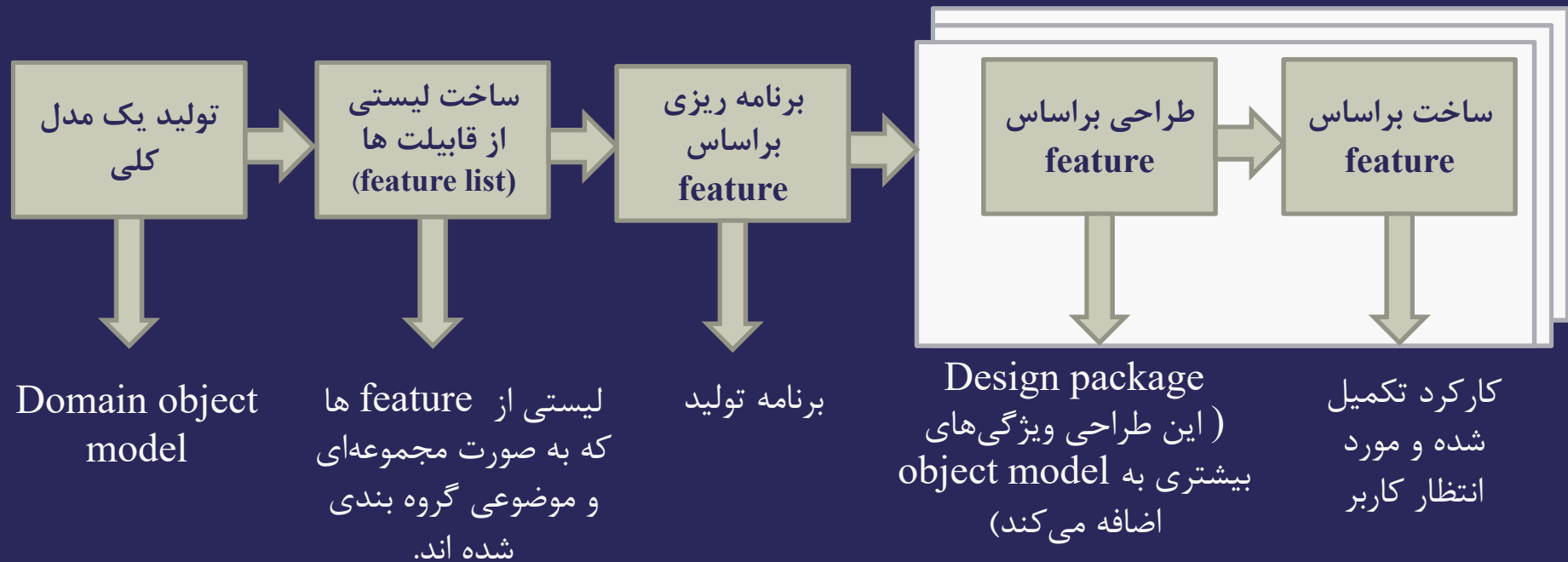
یک گروه قرار می دهد و به صورت زیر تعریف می گردد:

● $\langle \text{action} \rangle \langle \text{-ing} \rangle$ a(n) $\langle \text{object} \rangle$

● مثال: Making a product sale که مجموعه feature های بالا را در بر می گیرد.



FDD Processes





FDD Processes

➤ FDD بر فعالیتهای تضمین کیفیت با توجه به موارد زیر تاکید دارد:

- استراتژی تولید افزایشی
- Inspection طراحی و کد
- حسابرسی (audits)
- جمع آوری متریکها
- استفاده از الگوها (در آنالیز، طراحی و ساخت)

➤ در FDD برای مدیریت پروژه و زمانبندی، ۶ milestone در طی طراحی و پیادهسازی معرفی گردیده اند:

- design walkthrough, design, design inspection, code, code inspection, promote to build



منابع

- http://en.wikiversity.org/wiki/Crystal_Methods
- R. S. Pressman, *Software engineering: a practitioner's approach*. McGraw-Hill Higher Education, 2010.
- J. A. Highsmith, *Agile software development ecosystems*, vol. 13. Addison-Wesley Professional, 2002.