

مهندسی نرم افزار

مدرس: مسعود معاونی

منبع تدریس: کتاب مهندسی نرم افزار مهندس عین اله جعفرنژاد قمی و کتاب
مهندسی نرم افزار سجاد نقدعلی زنجانی

سرفصل‌های درس

- فصل اول: معرفی و آشنایی با ماهیت نرم‌افزار
- فصل دوم: فرایند
- فصل سوم: مهندسی سیستم و تحلیل نیازهای نرم‌افزار
- فصل چهارم: مدل‌های تحلیل
- فصل پنجم: مفاهیم طراحی
- فصل ششم: مدل‌های طراحی
- فصل هفتم: تعاریف و مفاهیم شی‌گرایی
- فصل هشتم: تحلیل و طراحی شی‌گرا
- فصل نهم: تعاریف و روش آزمون نرم‌افزار
- فصل دهم: راهبردهای آزمون نرم‌افزار
- فصل یازدهم: مدیریت پروژه و ارائه متریک‌های نرم‌افزاری
- فصل دوازدهم: تخمین و زمانبندی پروژه
- فصل سیزدهم: مباحث پیشرفته در مدیریت نرم‌افزار
- فصل چهاردهم: متدولوژی‌های نرم‌افزار

فصل دوم: فرایند

- مهندسی نرم افزار چیست؟

همانطور که در فصل قبل بیان شد، مهندسی نرم افزار فناوری است که در جهت تولید محصولی نرم افزاری به کاری می رود که در برگیرنده فرایند، مجموعه ای از روش ها و یک مجموعه از ابزارهاست.

- فرایند چیست؟

فرایند از مهمترین ارکان مهندسی نرم افزار است که مجموعه ای از فعالیت های اصلی (Framework Activity)، اقدامات (Action) و وظایف (Task) را جهت تهیه یک محصول نرم افزاری معرفی می کند.

مراحل ساخت نرم افزار

- برای ساخت هر نرم افزاری باید به سه مرحله **تعریف مسئله**، **توسعه راه حل** و **پشتیبانی** توجه داشت.
- در تعریف مسئله تمرکز اصلی بر روی برنامه ریزی، تعیین نیازهای سیستم و نرم افزار است.
- در مرحله توسعه، طراحی نرم افزار، تولید کد و همچنین تست نرم افزار مورد توجه قرار می گیرد تا محصول نرم افزاری آماده تحویل به مشتری شود.
- در نهایت در گام پشتیبانی، روشی برای کنترل تغییرات به هدف حفظ کارایی نرم افزار ارائه می شود.

مهندسی نرم افزار یک فناوری لایه‌ای

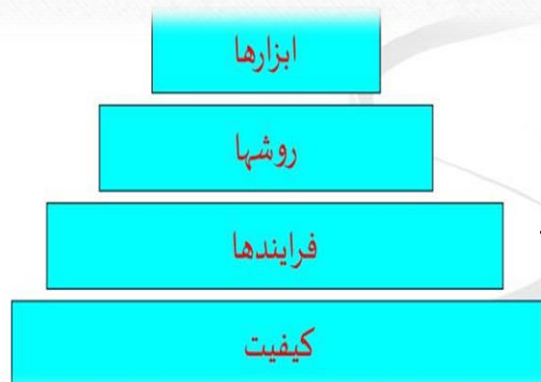
• به طور کلی مهندسی نرم افزار از چهار لایه ملاحظات کیفی، فرایند، روش‌ها و ابزارها تشکیل می‌شود.

• در پایین‌ترین لایه ملاحظات کیفی به دنبال ایجاد بهترین کیفیت هستیم

• در لایه فرایند روش‌ها و ابزارها تعیین می‌شوند.

• در لایه روش انجام وظایف از نظر فنی مشخص می‌شوند

• در بالاترین لایه ابزارهایی جهت پشتیبانی از لایه فرایند و روش استفاده می‌شوند.



فعالیت‌های اصلی در مهندسی نرم‌افزار

• دیدگاه ۱

- مهندسی سیستم
- تحلیل نیازمندی‌ها
- طراحی نرم افزار
- تولید کد
- آزمون
- پشتیبانی

• دیدگاه ۲

- ارتباط
- برنامه ریزی
- مدلسازی
- ساخت
- استقرار

فعالیت‌های چترگونه

• در تمامی مراحل اجرای فعالیت‌های قبلی، فعالیت‌های دیگری نیز به صورت پیوسته در حال اجرا هستند:

- کنترل و پیگیری پروژه‌های نرم‌افزاری
- مدیریت ریسک
- تضمین کنترل کیفیت نرم‌افزار
- بازبینی‌های فنی رسمی
- اندازه‌گیری
- مدیریت پیکربندی نرم‌افزار
- مدیریت قابلیت استفاده مجدد

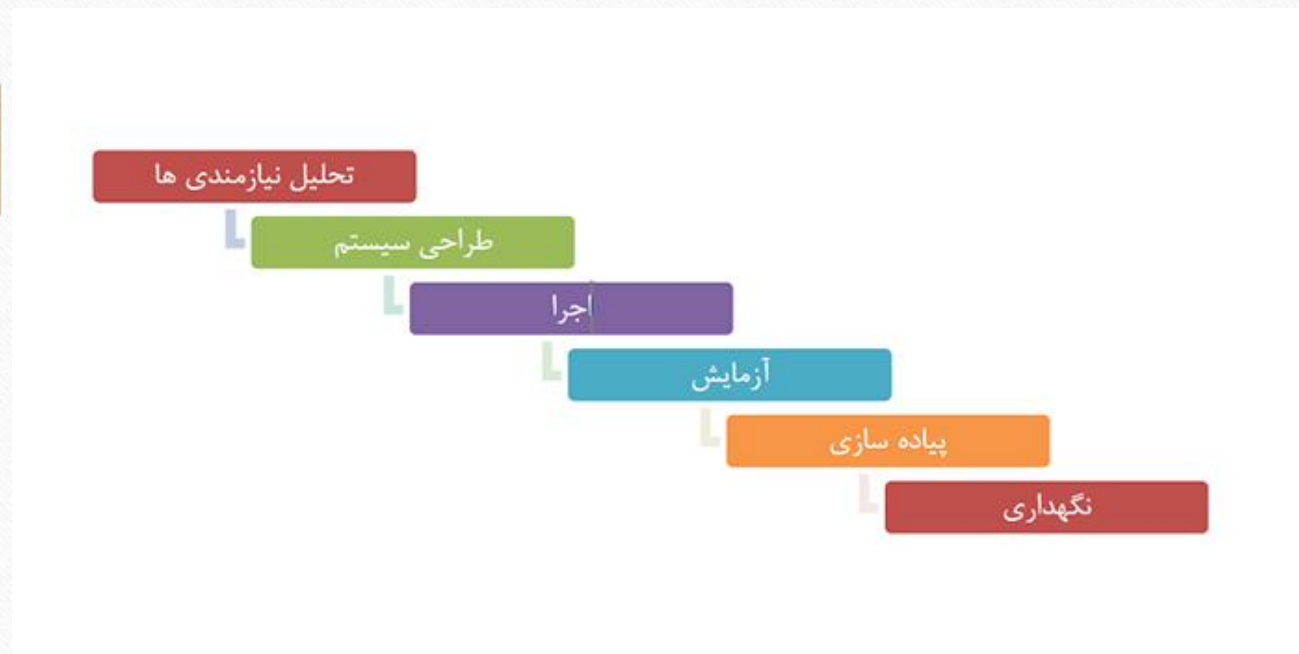
اجرای مهندسی نرم افزار

- به طور کلی مهندسی نرم افزار یک روال حل مشکل است. بنابراین باید ابتدا مسئله تعریف شود و سپس راه حلی برای آن ارائه شود.
- در گام تعریف با پرسیدن سوالات به دنبال درک مسئله هستیم
- در گام توسعه سعی می کنیم به ارزیابی راه حل های مختلف بپردازیم.
- در مرحله پشتیبانی با اعمال تغییرات راهکای بهینه تر را برای حل مسئله اصلی و مسائل جانبی ارائه می کنیم. پشتیبانی خود شامل اصلاح، تطابق، بهبود وضعیت و پیشگیری می شود.

آشنایی با انواع مدل‌های فرایند نرم‌افزار

- در طول دهه‌ها، مدل‌ها فرایند نرم‌افزار ساخت یافته و منسجم شده‌اند. براساس شرایط هر پروژه مهندسين نرم‌افزار از مدل خاصی استفاده می‌کنند.
 - هر مدل دارای نقاط قوت یا ضعفی است و تنها برای پروژه‌های خاصی می‌تواند مورد استفاده قرار گیرد.
- در ادامه لیستی از مدل‌های فرایند در مهندسی نرم‌افزار را مشاهده می‌کنیم:

۱- مدل ترتیبی-خطی (Linear Sequential Model)

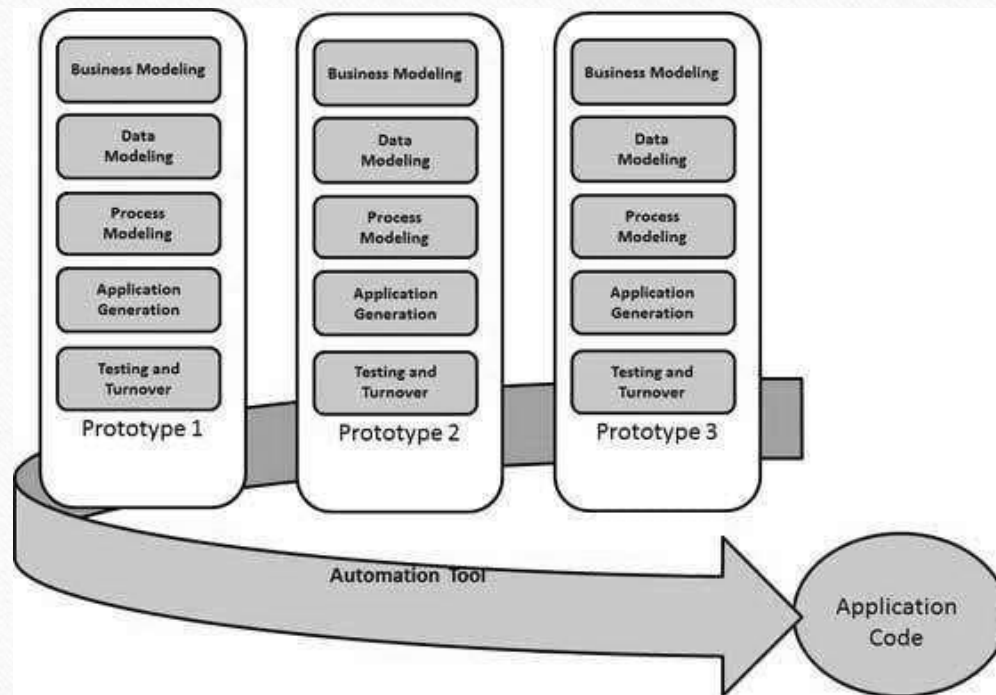


- مدل ترتیبی-خطی که با نام مدل‌های آبشاری (waterfall model) یا چرخه حیات کلاسیک (Classic life cycle) نیز شناخته می‌شود، یک مدل ساده و سنتی است که فعالیت‌های اصلی فرایند را به ترتیب و به شکل خطی اجرا کند.

ضعف‌های مدل ترتیبی-خطی

- نیازهای پروژه از ابتدا مشخص نیست و در طول پروژه مشخص می‌شود. همین مسئله باعث می‌شود تا در طول پروژه نیاز به تغییرات داشته باشیم.
- مدل آبشاری در پروژه‌های بزرگ با دو مشکل روبه‌رو است:
 - نسخه کاری تا انتهای پروژه در دسترس نیست
 - با گذشت زمان و طولانی شدن پروژه، احتمال تغییر در نیازهای شناسایی شده بیشتر می‌شود.
- مدل‌های خطی مشکل بلاک دارند. یعنی اعضای تیم باید منتظر انجام یک مرحله از کار توسط تیم دیگری شوند تا بتوانند کارشان را ادامه دهند.

۲- مدل ساخت سریع برنامه‌ها (Rapid Application Development)



- مدل RAD یک نسخه تطابق یافته و سرعت بالا از مدل آبشاری است. در این مدل ابتدا، مسئله به مسائل کوچکتر شکسته شده و هر تیم پروژه، مسئولیت یک یا چند بخش از مسائل پروژه را برعهده می‌گیرد. در طی آماده‌سازی هر بخش بیشترین تمرکز خود را بر افزایش سرعت توسعه می‌گذارد.
- در مدل RAD از روش‌های توسعه مبتنی بر مولفه (CBD) بهره می‌برد.

شرایط استفاده از مدل RAD

- یکی از مهمترین نکات جهت استفاده از مدل RAD، شفافیت در نیازهای پروژه است. هنگام استفاده از مدل RAD باید نیازهای پروژه از ابتدا مشخص باشد.
- در مدل RAD از روش تقسیم و غلبه استفاده می‌کنیم. اگر پروژه خاصیت پیمانه‌بندی کمی داشته باشد نمی‌توان انتظار کارایی داشت.
- در پروژه‌های بزرگ تعداد پیمانه‌ها (ماژول‌ها) زیادی است به همین دلیل به تیم‌های بیشتری نیاز دارید.
- وقتی پروژه با مسائل تکنیکال و فنی روبه‌رو شود تکنیک RAD کاربردی ندارد.
- موفقیت روش RAD وابسته به تعامل مناسب برنامه‌نویسان و مشتری است.

۳- مدل‌های تکاملی (Evolutionary Process)

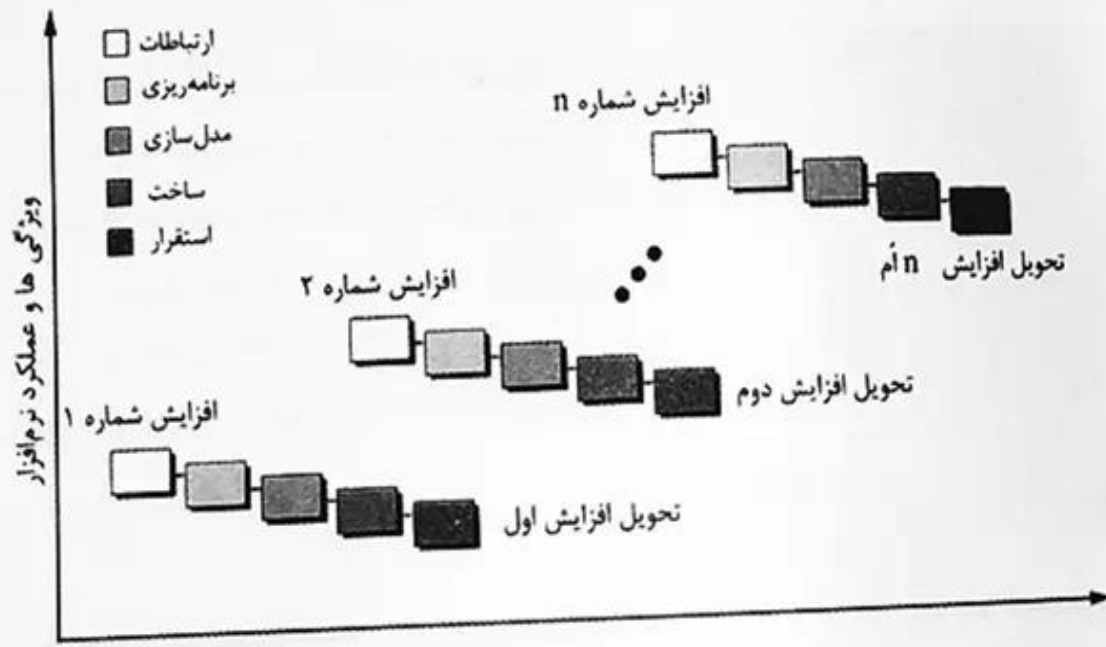
- مدل افزایشی (Incremental)
- مدل نمونه‌سازی (Prototyping)
- مدل پیچشی (Spiral)
- مدل حلزونی برنده-برنده (win-win spiral)

مدل افزایشی

- بسیاری از اوقات، شرکت‌های نرم‌افزاری تحت تاثیر فشارهای زمان‌بندی پروژه بوده و باید هر چه زودتر محصول نرم‌افزاری را به بازار عرضه کنند.

- از طرفی نیازها و محدوده پروژه نیز به وضوح مشخص نیست و امکان بروز تغییرات در طول پروژه وجود دارد.

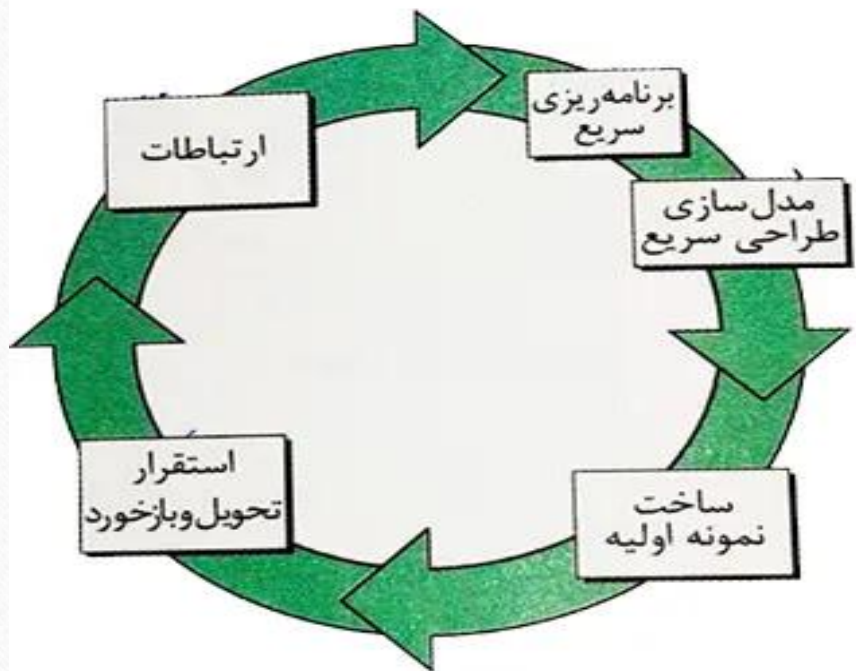
- مدل افزایشی، مدل ترتیبی-خطی را به صورت متناوب اجرا کرده و در هر تناوب محصولی کاربردی را به بازار عرضه می‌کند.



مزایای مدل افزایشی

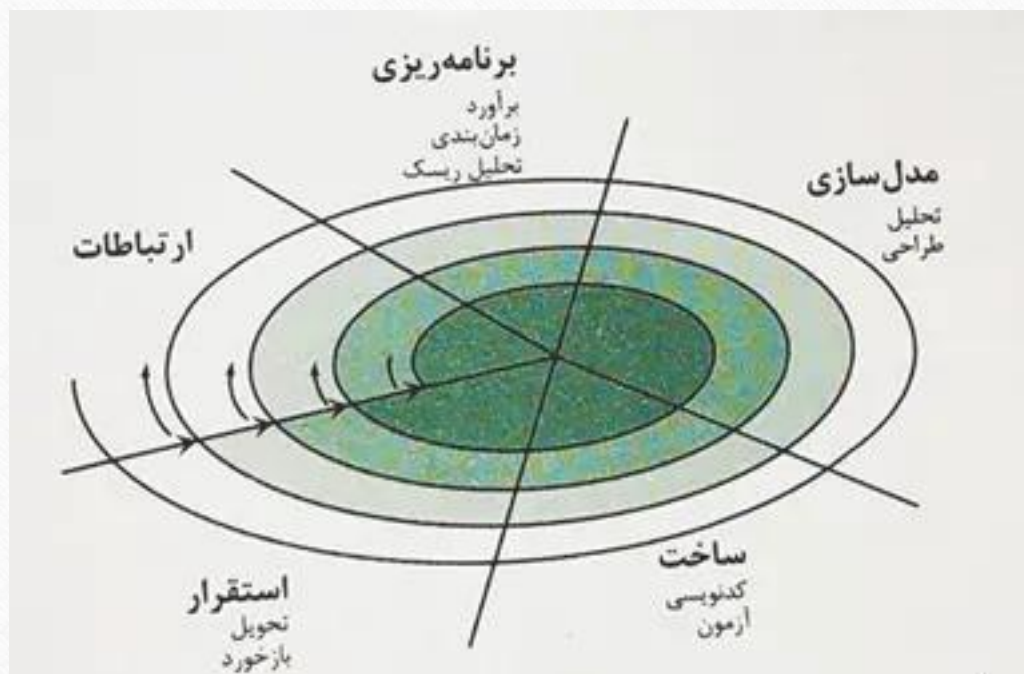
- مدل افزایشی در مقابل تغییرات منعطف بوده و امکان تکامل نرم افزار را فراهم می کند
- مدل افزایشی با اولویت بندی نیازهای مشتریان و ارائه نسخه های محدود، زمان تحویل پروژه را کاهش می دهد.
- مدل افزایشی با ارائه نسخه های محدود شده در هر افزایش، فشارهای زمان بندی را کاهش می دهد.
- زمانی که پروژه با کمبود نیروی انسانی مواجه باشد، مدل افزایشی مناسب است زیرا با نیروی انسانی کمتر می توان پروژه را جلو برد.

مدل نمونه سازی



- اغلب مشتریان یک مجموعه اهداف کلی را برای نرم افزار تعریف می کنند و در مورد جزئیات، کارکردها، وردی ها و خروجی ها نرم افزار مطالب دقیقی را تعریف نمی کنند. در مقابل مهندس نرم افزار نیز از محصول نهایی تصور درستی ندارد.
- در مدل نمونه سازی هدف ساخت یک نرم افزار کامل نیست، بلکه درک نیازهای مشتری و ارائه یک نمونه اولیه است.

مدل پیچشی



مدل پیچشی که با نام‌های حلزونی و مارپیچ نیز شناخته می‌شود یک مدل تکاملی است که به نوعی مدل آشنایی و مدل نمونه‌سازی را در بر می‌گیرد. در حقیقت مدل پیچشی مشابه مدل آشنایی زنجیره فعالیت‌ها را فعال کرده و مانند مدل نمونه‌سازی این زنجیره را به صورت گردشی اجرا می‌کند.

مدل حلزونی برنده-برنده

- در ذ این مدل که مشابه مدل پیچشی است، تیم تحلیل گر باید تلاش کند تا نیازهای مشتری را بهتر بشناسد و مشتری نیز باید جزئیات را در اختیار تحلیل گران قرار دهد.
- این مدل براساس سه اصل کلیدی زیر شکل می گیرد:
 - شناسایی سیستم و ذی نفعان آن
 - تعیین شرایط برد ذی نفع ها
 - بررسی و مذاکره در مورد شرایط برد ذی نفع ها و ایجاد مصالحه در بین شرایط

دیگر مدل‌های فرایند

- **مدل توسعه همروند:** اکثر سازمان‌های نرم‌افزاری در یک بازه زمانی چندین پروژه را در دست توسعه دارند. با استفاده از مدل همروند می‌توان تجاربی که از پروژه‌های قبلی داریم را بر روی پروژه‌های جدید لحاظ کنیم.
- **مدل توسعه مبتنی بر مولفه:** مولفه‌ها بسته‌های آماده هستند که شامل چندین تابع و ماژول می‌شوند. با استفاده از مولفه‌های از پیش آماده شده سرعت و هزینه انجام پروژه کاهش می‌یابد.
- **مدل‌های شیوه رسمی:** در مدل‌های فرایندی ممکن است مهندسین نرم‌افزار در تحلیل نیازها یا روش اجرا دچار اختلاف نظر شوند. سوبرداشته‌ها از جمله موارد اصلی در شکست نرم‌افزار است. در این حالت سعی می‌شود تا از استانداردهای رسمی برای طراحی فرایند استفاده شود.

سوالات پایان فصل ۲

- مهندسی نرم افزار را تعریف کنید؟ منظور از فرایند چیست؟
- فعالیت های اصلی در مهندسی نرم افزار را نام ببرید؟
- مدل ترتیبی-خطی را توضیح داده و معایب آن را نام ببرید؟
- در چه شرایطی از مدل ساخت سریع یا همان RAD استفاده می کنیم؟
- مزایای کلیدی مدل افزایشی چیست؟
- مدل نمونه سازی را توضیح داده و کاربرد آن را بیان کنید؟
- مزیت کلیدی استفاده از مدل توسعه مبتنی بر مولفه چیست؟

ادامه سوالات

- کدام یک از گزینه‌های زیر جز لایه‌های مهندسی نرم‌افزار نیست؟
 - فرایند
 - ابزار
 - روش‌ها
 - محصول
- مدل حلزونی برنده-برنده به کدام ویژگی بیشتر توجه دارد:
 - ارتباط با مشتری
 - مهندسی و طراحی
- ساخت و نصب
- تحلیل ریسک
- با پروژه‌ای مواجه‌ایم که نیازهای آن مشخص نیست، کاربر نمی‌داند چه می‌خواهد و تیم توسعه هم درباره محصول نهایی مطمئن نیست. کدام گزینه مناسب‌تر است؟
 - مدل حلزونی
 - مدل افزایشی
 - مدل نمونه‌سازی
 - مدل مبتنی بر مولفه یا CBD