

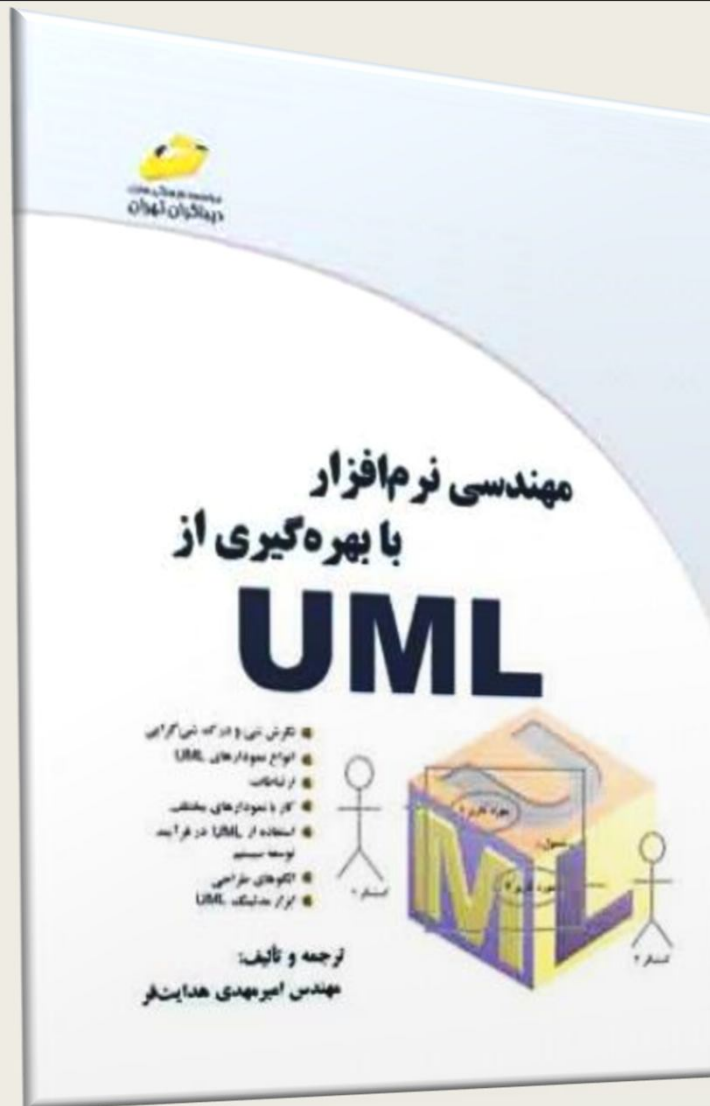


دانشگاه جامع علمی کاربردی

مهندسی نرم افزار

مدرس: یکماز

منبع درسی



■ عنوان:

مهندسی نرم افزار با بهره گیری از UML

■ نویسنده:

امیر مهدی هدایتی

■ انتشارات:

دنیای گران تهران

■ منبع اصلی:

Software Engineering

Roger S. Pressman

McGraw-Hill

بارم بندی

- پایان ترم: ۱۴ نمره
- حل تمرینات و مشارکت در مباحث کلاسی: ۶ نمره
- حضور در کلاس: ۲- تا ۲ نمره

سرفصل آموزشی

- محصول (نرم افزار)
- فرآیند توسعه نرم افزار
- مهندسی سیستم
- اصول و قواعد تحلیل نیاز
- مدل سازی تحلیل نیازهای نرم افزار
- اصول و قواعد طراحی نرم افزار
- طراحی معماری نرم افزار
- طراحی واسط کاربر
- طراحی در سطح مولفه
- اصول و قواعد تحلیل شیء گرا

سرفصل آموزشی

- مدل سازی تحلیل شیء گرا
- طراحی شیء گرا
- آزمون نرم افزار و راهبردها
- مفاهیم مدیریت پروژه های نرم افزاری
- معیارهای اندازه گیری و سنجش نرم افزار
- برنامه ریزی پروژه های نرم افزاری
- تحلیل و مدیریت ریسک
- زمان بندی، نظارت و کنترل پروژه
- اطمینان مرغوبیت نرم افزار
- مدیریت پیکربندی نرم افزار

فصل اول

محصول (نرم افزار)

تعریف مهندسی نرم افزار

■ از دیدگاه Pressman

■ به مجموعه ای از فن آوری ها، روش ها و ابزارهای مبتنی بر اصول مهندسی که در توسعه نرم افزار استفاده می شود، مهندسی نرم افزار گفته می شود.

■ از دیدگاه Sommerville

■ مهندسی نرم افزار در دهه ۱۹۶۰ در کنفرانسی با نام **بحران نرم افزار** مطرح می گردد.

■ مهندسی نرم افزار در ارتباط با توسعه سیستم های نرم افزاری به شکل فعالیت گروهی است و از قواعد مهندسی استفاده می کند و شامل جنبه های فنی و غیر فنی است.

تعریف مهندسی نرم افزار

■ از دیدگاه Bauer

■ مهندسی نرم افزار عبارت از توسعه و استقرار نرم افزار با استفاده از اصول دقیق مهندسی برای دستیابی به نرم افزاری با صرفه اقتصادی است دو شرط قابل اعتماد (Reliable) و کارا (Efficient) را دارد

نکته: مهندسی نرم افزار هم ارز با برنامه نویسی نیست، بلکه برنامه نویسی فقط جزئی از مهندسی نرم افزار است.

مشکلات کنونی نرم افزار

- عدم تطابق نرم افزار با نیازهای واقعی مشتری و بازار (تغییرات) -
- عدم امکان رقابت و استفاده کارای نرم افزار از سخت افزار با توجه به پیشرفت سریع سخت افزار (فن آوری)
- عمومی شدن کاربرد رایانه در جامعه و عدم امکان تغییر و تعویض نرم افزار در مدت زمان کوتاه (کاربرد لحظه ای)
- فشار عمومی و زیاد برای تولید نرم افزاری قابل اعتماد و دارای کیفیت مناسب (آزمون نرم افزار)
- قدرت محدود در پشتیبانی و نگهداری نرم افزارهای موجود (قابلیت توسعه)

تکامل نرم افزار - طرح سؤال در برابر مشکلات

■ چرا توسعه نرم افزار طولانی است؟

❖ عامل مهم در طولانی شدن توسعه نرم افزار ها، عدم امکان اندازه گیری مستقیم و عدم استفاده از مؤلفه های آماده و آزمایش شده در پیاده سازی است. راهکار استفاده از روش توسعه مبتنی بر مؤلفه (CBD) تاحدی می تواند در رفع این مشکل کمک کند.

■ چرا هزینه توسعه نرم افزار بالا است

❖ به علت تغییرات محیطی، فنی و طولانی شدن زمان توسعه نرم افزار، هزینه ها افزایش می یابد

تکامل نرم افزار – طرح سؤال در برابر مشکلات

■ چگونه می توان خطاهای نرم افزار را پیش از تحویل شناسایی کرد؟

❖ در مراحل تحلیل و طراحی نرم افزار، بازبینی های مستمر انجام و کد تولید شده، به حد کفایت مورد آزمایش قرار گیرد

■ چرا در اندازه گیری پیشرفت پروژه نرم افزاری مشکل وجود دارد؟

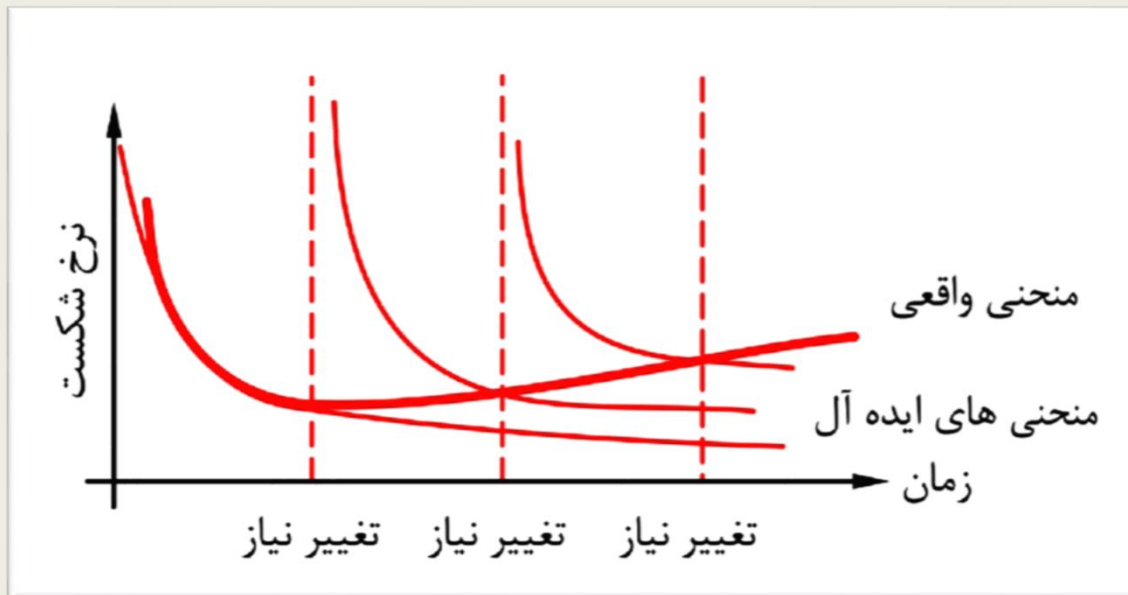
❖ چون معیارهای دقیق و خاصی برای سنجش نرم افزار وجود ندارد و این معیارها به پارامترهای بسیاری از تحلیل، طراحی و پیاده سازی وابسته اند

ویژگی های نرم افزار

- نرم افزار یک محصول منطقی است و یک محصول فیزیکی نیست.
- نرم افزارها بیشتر بر اساس نیاز مشتریان ساخته می شوند.
- نرم افزار دورانداختنی نیست.

ویژگی های نرم افزار

- در شکل زیر به وضوح دیده می شود که منحنی عمر مفید نرم افزار محدب و دارای نرخ شکست صعودی با شیب بسیار کم نسبت به منحنی ایده آل در طول زمان است که این رشد نرخ شکست ناشی از تغییر نیازها است.



شکل ۱-۱: منحنی های شکست نرم افزار

کاربردهای نرم افزار

■ تعیین ماهیت کاربردی یک نرم افزار

■ محتوای اطلاعات

■ معنی و شکل اطلاعات ورودی و خروجی

■ قطعیت اطلاعاتی

■ پیش بینی ترتیب و زمان بندی اطلاعات

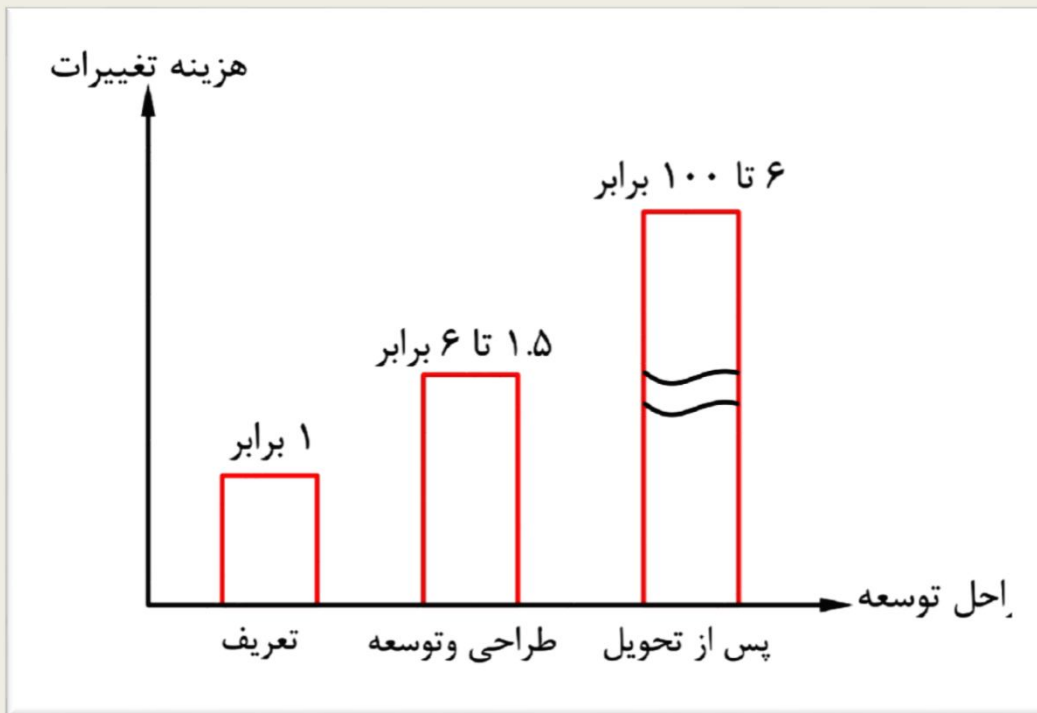
کاربردهای نرم افزار

■ گروه های کاربردی

- نرم افزارهای سیستمی
- نرم افزارهای زمان حقیقی
- نرم افزارهای تجاری
- نرم افزارهای مهندسی و علمی
- نرم افزارهای توکار و تعبیه شده
- نرم افزارهای مبتنی بر رایانه های شخصی
- نرم افزارهای مبتنی بر وب
- نرم افزارهای هوش مصنوعی و شبکه های عصبی و ...

هزینه های ناشی از تغییرات در چرخه توسعه نرم افزار

- برآوردهای زیر از تعداد زیادی نرم افزار در چرخه توسعه نرم افزار انجام شده است که نشان دهنده شدت افزایش هزینه تغییرات پس از تحویل نرم افزار است.



شکل ۱-۲ هزینه تغییرات در چرخه توسعه نرم افزار