

مهندسی نرم افزار

شامل:

- متن کامل و طبقه بندی شده درس بر اساس سرفصل های مصوب وزارت علوم
- تست های طبقه بندی شده آزمون های کارشناسی ارشد مهندسی IT با حل تشریحی
- در حدود ۱۰۰ تست تالیفی طبقه بندی شده با پاسخ تشریحی ده ها مسئله تالیفی برای خود آزمایی

مؤلف: سجاد نقدعلی زنجانی

سرشناسه	نقدعلی زنجانى، سجاد، ۱۳۶۳-
عنوان و نام پدیدآور	مهندسی نرم افزار: قابل استفاده دانشجویان ... / مولف سجاد نقدعلی زنجانى؛ [به سفارش مؤسسه آموزش عالی آزاد پارسه].
مشخصات نشر	تهران: حرکت نو، ۱۳۹۲.
مشخصات ظاهری	۳۳۰ ص.: مصور.
شابک	۹۷۸-۶۰۰-۷۰۳۵-۱۶-۰
وضعیت فهرست نویسی	فیب
موضوع	دانشگاه ها و مدارس عالی -- ایران -- آزمون ها
موضوع	نرم افزار -- مهندسی -- راهنمای آموزشی (عالی)
موضوع	نرم افزار -- مهندسی -- آزمون ها و تمرین ها (عالی)
موضوع	آزمون دوره های تحصیلات تکمیلی -- ایران
شناخه افزوده	مؤسسه آموزش عالی آزاد پارسه
رده بندی کنگره	۱۳۹۲ م ۹ / ن ۳۵۳ LB
رده بندی دیویی	۳۷۸/۱۶۶۴
شماره کتابشناسی ملی	۳۲۲۲۱۰۵

عنوان کتاب	مهندسی نرم افزار
مؤلف	سجاد نقدعلی زنجانى
با مدیریت	ابوالفضل طرقي حقيقت
ناشر	حرکت نو
نوبت چاپ	اول - تابستان ۱۳۹۲
شمارگان	۱۵۰۰
طرح جلد	مؤسسه آموزش عالی آزاد پارسه
لینوگرافی، چاپ و صحافی	عمران
قیمت	۲۷۰۰۰ تومان
شابک	۹۷۸-۶۰۰-۷۰۳۵-۱۶-۰

تمام حقوق مادی و معنوی این اثر متعلق به مؤسسه آموزش عالی آزاد پارسه است.
توجه: به موجب قانون حمایت حقوق مؤلفان و مصنفان و هنرمندان مصوب ۱۳۴۸/۱۰/۱۱ و همچنین قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی مصوب سال ۱۳۵۰ تمام حقوق این اثر به هر نحو برای مؤسسه آموزش عالی آزاد پارسه محفوظ است. نشر و پخش تمام یا قسمتی از این کتاب بدون اجازه، باعث پیگرد قانونی خواهد شد.

یادداشت ناشر

خرد دست گیرد به هر دو سرای
تو بسی چشم شادان جهان نسیری
به گیتی پیوی و به هر کس بگوی
از آموختن یک زمان نغسوی
فردوسی

خرد رهنمای و خرد دلگشای
خرد چشم جان است چون بنگری
به گفتار ار داندگان راه جوی
ز هر دانش چون سخن بشوی

در دیایی زندگی می کنیم که تکنولوژی به پیش پا افتاده ترین جنبه های زند گیمان سرک می کشد و همه مرزهایش را درنور دیده، در این میانه دسترسی به متون و منابع نگارشی با یک کلیک میسر می شود و هر فرد می تواند صاحب کتابخانه ای بزرگ باشد، اما با تمام این تفاسیر، آنچه شگفت انگیز است مثل انسان ها به داشتن و خواندن کتاب های واقعی است، کتاب هایی از جنس کاغذ و جوهر که سال ها است از نسلی به نسل دیگر در سفرند و چرخ های سترگ تکنولوژی را به حرکت در می آورند. شاید این خاصیت کاغذ و جوهر است که گویی با علم و دانش پیمانی ناگسستی بسته است، اتاق اساتید بزرگ دانشگاه را که نگاه می کنی نخستین و برجسته ترین نکته ای که نظرت را جلب می کند کتابخانه های غنی و کتب منبع آنهاست که همگی شاید یادگار روزگار دانشجویی و علم آموزشان بوده است.

مؤسسه آموزش عالی آزاد پارسه نیز آرزو دارد کتاب هایش همواره همراه شما باشند از این روست که حداکثر تلاش خود را کرده تا این کتاب ها علاوه بر دربرگیری همه نکاتی که برای یک دانشجوی کنکوری مورد نیاز است با بیانی روان و قابل فهم کلیه مطالب و مباحث آکادمیک را شرح و بسط دهد تا خواننده را از رجوع به کتاب های مشابه بی نیاز کند و سالها در کتابخانه های شما به عنوان مرجعی ماندگار و جامع حضور داشته باشد.

پارسه دغدغه ای دیگر نیز دارد، امروز که اکثریت نسل جوان جامعه ما دارای تحصیلات عالی هستند، جای تفکر و تأمل بسیار است که سرانه کتاب و کتابخوانی هر ایرانی از جایگاه نازلی در میان دیگر کشورهای دنیا برخوردار باشد. این حقیقت تلخ برای شما نمایانگر وقتی نگاهی به پیشینه فرهنگ و هنر در این مرز و بوم ببینیم که هیچ کس نمی تواند منکر برجستگی و شکوهِش باشد. شاید وظیفه هر انسان دلسوز و نهاد آموزشی و فرهنگی این است که در راه اعتلای مجدد این فرهنگ گام بردارد، چرا که این قدم ها هر چند کوچک هم که باشد می تواند مقدم های گردد برای خیزی بلند و سرافرازانه به جایگاه مرتفع اسلاف بزرگ و ماندگارمان. پارسه نیز بر خود بایسته می داند با رویکردهای فرهنگی، مانند اهدا کتاب و تجهیز کتابخانه ها به سهم خود در پیشبرد این جریان همت گمارد و امیدوار است در آینده نزدیک شاهد شکوفایی دوباره این فرهنگ فاخر در کشورمان باشد.

مؤسسه آموزش عالی آزاد پارسه از شما تقاضا دارد تا نظرات و پیشنهادات خود را در مورد کتاب های این مؤسسه به پست الکترونیکی به آدرس CRM@parseh.ac.ir ارسال کنید تا هر روز و با یکدیگر به سمت بهتر شدن پیش رویم.

الا ای پیر دانای قلم در دست
 تو ای مجنون که لیلایت بُود هستی
 تو ای ساقی که خود بیش از همه مستی
 تو ای آنکس که نام خویش را هم برده‌ای از یاد
 چگانه اینچنین آیا رها گشتی، شدی چون باد؟

دکتر حقیقت

به نام آن عشق جاودان بیکران که نام و نشانی ندارد و مستی و رهایی از هر قید و بند و آرزو، شرط فنا شدن در اوست.

تردیدی نیست که نرم‌افزار در سالیان اخیر نقش چشم‌گیری در ارتقاء تکنولوژی بشر داشته است. عجب آن که این تأثیرگذاری، طیف وسیعی از تکنولوژی‌های بشری را در بر گرفته، به طوری که امروزه به سختی می‌توان فناوری را یافت که با ابزارهای نرم‌افزاری عجین نشده باشد. در این راستا تمایل به استفاده از نرم‌افزارهای جامع‌تر و مناسب‌تر در هر تکنولوژی، باعث شده فرآیند توسعه نرم‌افزار به یکی از جدی‌ترین حوزه‌های پروژه‌های فنی تبدیل شود؛ فرآیندی که بدون توجه به فعالیت‌های مدیریتی و فنی به سرانجام نرسد.

علم مهندسی نرم‌افزار، فرآیند توسعه نرم‌افزار را تشریح می‌کند؛ نقشی انجام کار، روش‌های گوناگون توسعه و ابزارهای مختلفی که در این مسیر استفاده می‌شوند، عمده نقاط تمرکز این علم هستند. در کتاب حاضر تلاش شده تمامی حوزه‌های اصلی مهندسی نرم‌افزار مورد رسیدگی قرار بگیرند. در این کتاب نه تنها بر فعالیت‌های فنی حاضر در فرآیند توسعه، تأکید ویژه‌ای شده است، بلکه رویکردها و روش‌های مدیریتی پروژه‌های نرم‌افزاری نیز مورد بررسی قرار گرفته است. فصول این کتاب بر اساس ترتیب اجرای مراحل فنی نرم‌افزار (تعیین مدل فرآیندی، تحلیل، طراحی، پیاده‌سازی و آزمون) تدوین شده است؛ البته در چهار فصل انتهایی این کتاب نیز فعالیت‌های مدیریتی همچون تشکیل تیم‌های نرم‌افزاری، تخمین و زمانبندی پروژه، تحلیل ریسک، تضمین کیفیت نرم‌افزار و مدیریت تغییرات مورد رسیدگی قرار گرفته است. اگر قرار باشد تنها بر یکی از امتیازات این کتاب تأکید شود؛ بیان جامع مطالب در ساختاری منسجم و پیوسته است به طوری که مطالب در ساده‌ترین و کوتاه‌ترین شکل ممکن به خواننده انتقال می‌یابد. در این کتاب از بیان مطالب جزیره‌ای (غیر مرتبط) و ترجمه‌های غیر ملموس که تنها باعث آزرده‌گی و دل‌زدگی خواننده می‌شود به جد اجتناب شده است.

اگرچه یکی از اهداف این کتاب آماده‌سازی خواننده در کنکور کارشناسی ارشد (و تا حدی دکترا) است، اما این امر باعث نشده نویسنده از جنبه‌های عملی و اجرایی مطالب عنوان شده غافل شود و تنها مجموعه‌ای از نکات کنکوری را ارائه دهد؛ مثال‌های عملیاتی که در رابطه با هر مبحث مطرح شده و زنجیره‌ای از مطالب مرتبط باعث شده مخاطبان این کتاب نه تنها داوطلبین

کنکور، بلکه مهندسين و متخصصين نرم افزار، دانشجويان، فارغ التحصيلان و در مجموع همه ي افرادی باشند که می خواهند بدانند چطور می توان نرم افزاری با کیفیت را مهندسی کرد. در این کتاب مجموعه ای از سوالات چهارگزینه ای کنکورهای سال های گذشته فراهم آمده تا خوانندگان بتوانند بعد از اتمام هر فصل، آموخته های خود را محک بزنند. همچنین با هدف تکمیل این ارزیابی، مجموعه ای از سوالات چهارگزینه ای در آخر هر فصل طراحی شده، تا هم بر حوزه های مهم آن فصل تأکید بیش تری شود و هم اگر جنبه هایی از آن فصل در کنکورهای گذشته مورد توجه قرار نگرفته به آن مباحث نیز پرداخته شود. همچنین یکی دیگر از مواردی که در کتاب حاضر مورد توجه بوده، سوالات خودآزمایی است که ضمن مرور مطالبی علمی فصل، مباحث و عناوین تکمیلی مرتبط با فصل را مطرح می کند تا خوانندگان علاقه مند بتوانند در صورت تمایل به کسب اطلاعات بیشتر، هر یک از این عناوین را مورد بررسی و پژوهش بیشتر قرار دهند.

در این جا شایسته است از دوست خوبم مهندس حمیدرضا کدخدایی که به عنوان بازنگرنده ی نهایی، در فرآیند تالیف این کتاب فعالیت داشته است، یاد شود. ایشان با حوصله و دلسوزی فراوان، ضمن بازبینی کتاب، از طریق یک نگاه علمی، حس و نظر خواننده را مشخص می نمودند. من به جد جایگاه کیفی این کتاب را در گروی تلاش ها و مساعدت وی دانسته و از صمیم قلب از ایشان تشکر می کنم.

در پایان دوست دارم در یک بیان صمیمانه، به خوانندگان محترم ابراز کنم که در نوشتن این کتاب چیزی به جز عشق و خدمت را لحاظ نکرده ام. تمام فکرم آن بوده که هر آنچه در توان دارم را در این مسیر به کار گیرم؛ در واقع اگر بتوان یادگیری مباحث مهندسی نرم افزار را به یک سفر دریایی تمثیل کرد، هدف اصلی من ساخت یک کشتی مطمئن بوده که مسافران آن بتوانند بدون نگرانی از مشکلات راه، سفر خود را با لذت از مناظر حاضر در مسیر، به پایان برند. البته می دانیم با وجود این تلاش ها، کتاب حاضر حاوی کاستی هایی است که با یاری شما بزرگواران این کاستی ها را نیز بر طرف خواهیم نمود.

این کتاب با مدیریت دکتر "ابولفضل طرقي حقیقت" تهیه و تدوین شده و به تمامی دانشجويان و علاقه مندان تقدیم می شود و از همه عزیزان تقاضا می شود که پس از مطالعه کتاب نظرات ارزشمند خود را از طریق آدرس پستی الکترونیکی at_haghighat@yahoo.com یا آدرس سایت www.at-haghighat.com به مولف ارسال نمایند تا در بهبود سطح کیفی ویراست های بعدی کتاب مثرمتر واقع شود.

فهرست

فصل اول

- ۱-۱ مقدمه
- ۲-۱ نرم افزار، محصولی متفاوت
- ۳-۱ کاربردهای نرم افزار
- ۴-۱ باورهای قدیمی و علم اشتباه!
- ۱-۴-۱ اشتباهات مدیران
- ۲-۴-۱ اشتباهات مشتری
- ۳-۴-۱ اشتباهات برنامه‌نویسان
- تست‌های فصل اول
- پاسخ تست‌های فصل اول
- خودآزمایی

فصل دوم

- ۱-۲ مقدمه
- ۲-۲ مهندسی نرم افزار یک فناوری لایه‌ای
- ۱-۲-۲ ابزارهای CASE
- ۳-۲ فرآیند نرم افزار
- ۴-۲ فعالیت‌های اصلی (Framework Activity)
- ۱-۴-۲ فعالیت‌های اصلی عمومی
- ۲-۴-۲ فعالیت‌های چترگونه (umbrella activity)
- ۵-۲ مهندسی نرم افزار در یک دید کلی
- ۱-۵-۲ تعریف
- ۲-۵-۲ توسعه
- ۳-۵-۲ مرحله پشتیبانی
- ۶-۲ مدل بلوغ توانایی (CMM)
- ۷-۲ مدل‌های فرآیند نرم افزار
- ۱-۷-۲ مدل ترتیبی - خطی (linear sequential model)

۱۹	۲-۷-۲ مدل ساخت سریع برنامه‌ها (RAD) (rapid application development)
۲۰	۳-۷-۲ مدل‌های تکاملی (evolutionary process)
۲۳	۴-۷-۲ دیگر مدل‌های فرآیند
۲۵	۸-۲ مباحث تکمیلی
۲۷	تست‌های فصل دوم
۳۱	پاسخ تست‌های فصل دوم
۳۵	خودآزمایی
	فصل سوم
۳۷	۱-۳ مقدمه
۳۸	۲-۳ مهندسی سیستم
۳۸	۱-۲-۳ سیستم کامپیوتری
۳۹	۲-۲-۳ دو جنبه‌ی مختلف از مهندسی سیستم
۴۱	۳-۲-۳ مهندسی نیازمندی‌ها
۴۳	۴-۲-۳ مدل‌سازی سیستم
۴۵	۳-۳ تحلیل نیازمندی‌ها
۴۶	۱-۳-۳ مراحل تحلیل نیاز
۴۷	۲-۳-۳ تکنیک‌های تحلیل نیاز
۴۸	۳-۳-۳ اصول تحلیل نیاز
۵۰	۴-۳-۳ مورد کاربرد (Use Case)
۵۳	تست‌های فصل سوم
۵۵	پاسخ تست‌های فصل سوم
۵۷	خودآزمایی
	فصل چهارم
۵۹	۱-۴ مقدمه
۶۰	۲-۴ مدل‌سازی داده‌ها
۶۱	۱-۲-۴ اشیاء داده‌ای و صفات آن‌ها
۶۲	۲-۲-۴ روابط بین اشیاء داده‌ای
۶۳	۳-۴ مدل کارکردی (functional model)
۶۳	۱-۳-۴ اجزای نمودار DFD
۶۴	۲-۳-۴ سطوح مختلف تجرید در نمودار DFD
۶۶	۳-۳-۴ نکات رسم نمودار DFD
۶۶	۴-۳-۴ مدل‌های منطقی و فیزیکی در DFD

۶۷	۴-۴ مدل رفتاری (behavioral model)
۶۷	۱-۴-۴ مدل وارد (Ward)
۶۸	۲-۴-۴ مدل هاتلی (Hatley)
۷۰	۳-۴-۴ ارتباط نمودار CFD با نمودار DFD
۷۱	تست‌های فصل چهارم
۷۳	پاسخ تست‌های فصل چهارم
۷۵	خودآزمایی
	فصل پنجم
۷۷	۱-۵ مقدمه
۷۸	۲-۵ ابعاد طراحی
۷۹	۳-۵ اهداف، اصول و نکات طراحی
۸۰	۴-۵ مفاهیم طراحی (design concepts)
۸۰	۱-۴-۵ تجزیه (abstraction)
۸۰	۲-۴-۵ پالایش (refinement)
۸۱	۳-۴-۵ پیمانه‌سازی (modularity)
۸۲	۴-۴-۵ معماری نرم‌افزار
۸۲	۵-۴-۵ سلسله مراتب کنترل (control hierarchy)
۸۳	۶-۴-۵ ساختمان داده‌ها و رویه‌ی نرم‌افزار
۸۴	۷-۴-۵ پنهان‌سازی اطلاعات (information hiding)
۸۴	۸-۴-۵ استقلال عملکردی (Functional independence)
۸۸	۵-۵ چند نکته در طراحی پیمانه‌ای
۹۱	تست‌های فصل پنجم
۹۵	پاسخ تست‌های فصل پنجم
۹۷	خودآزمایی
	فصل ششم
۹۹	۱-۶ مقدمه
۱۰۰	۲-۶ معماری نرم‌افزار (software architecture)
۱۰۰	۱-۲-۶ طراحی داده‌ها
۱۰۰	۲-۲-۶ سبک‌های مختلف معماری
۱۰۶	۳-۲-۶ ارزیابی معیارهای جایگزین (assessing alternative architecture)
۱۰۹	۴-۲-۶ زبان‌های توصیف معماری (architectural description Languages)
۱۰۹	۵-۲-۶ نگاشت نیازمندی‌های نرم‌افزار به سبک‌های معماری

۱۱۴	۳-۶ واسطه (interface)
۱۱۴	۱-۳-۶ قوانین طلایی
۱۱۵	۲-۳-۶ تحلیل و طراحی واسطه کاربر
۱۱۹	۴-۶ طراحی تفصیلی (طراحی در سطح اجزاء)
۱۲۰	۱-۴-۶ فلو چارت
۱۲۱	۲-۴-۶ نمودار جمع‌های
۱۲۲	۳-۴-۶ جدول تصمیم‌گیری (decision table)
۱۲۲	۴-۴-۶ زبان‌های طراحی برنامه (Program Design Language)
۱۲۵	تست‌های فصل ششم
۱۲۹	پاسخ تست‌های فصل ششم
۱۳۱	خودآزمایی

فصل هفتم

۱۳۳	۱-۷ مقدمه
۱۳۴	۲-۷ اشیاء (objects)
۱۳۵	۳-۷ کلاس‌ها
۱۳۶	۴-۷ ویژگی‌های منحصربه‌فرد شی‌گرایی
۱۳۶	۱-۴-۷ بسته‌بندی (encapsulation)
۱۳۷	۲-۴-۷ ارث‌بری (inheritance)
۱۴۰	۳-۴-۷ چندریختی (polymorphism)
۱۴۲	۵-۷ شناخت عناصر مدل شی
۱۴۳	۶-۷ مهندسی دامنه (Domain engineering)
۱۴۵	تست‌های فصل هفتم
۱۴۷	پاسخ تست‌های فصل هفتم
۱۴۹	خودآزمایی

فصل هشتم

۱۵۱	۱-۸ مقدمه
۱۵۲	۲-۸ شناخت موارد کاربرد (use cases)
۱۵۲	۱-۲-۸ رابطه شامل بودن (include)
۱۵۲	۲-۲-۸ رابطه‌ی گسترش (extend)
۱۵۳	۳-۲-۸ رابطه‌ی تعمیم (generalization)
۱۵۳	۳-۸ شناخت کلاس‌ها، اشیاء و صفات و عملیات آن‌ها
۱۵۴	۱-۳-۸ مدل‌سازی کلاس، مسئولیت - همکاری (CRC)

۱۵۵	۲-۳-۸ تحلیل بسته (package analysis)
۱۵۶	۴-۸ کلاس دیاگرام
۱۵۶	۱-۴-۸ معرفی کلاس دیاگرام
۱۵۷	۲-۴-۸ رابطه‌ی انجمنی (association relationship)
۱۵۸	۳-۴-۸ رابطه‌ی تعمیم/تخصیص (generalization/specialization)
۱۵۸	۴-۴-۸ رابطه‌ی تجميع (aggregation)
۱۵۹	۵-۴-۸ اصل ترکیب (composition relationship)
۱۵۹	۶-۴-۸ رابطه‌ی وابستگی (dependency relationship)
۱۶۰	۵-۸ رسم مدل‌های رفتاری
۱۶۰	۱-۵-۸ نمودار تالی (sequence diagram)
۱۶۱	۲-۵-۸ نمودار حالت (state diagram)
۱۶۲	۳-۵-۸ دیاگرام ارتباط (communication diagram)
۱۶۳	۶-۸ مدل‌سازی جزئیات پیاده‌سازی
۱۶۳	۱-۶-۸ نمودار مؤلفه (component)
۱۶۴	۲-۶-۸ نمودار فعالیت (activity diagram)
۱۶۵	۷-۸ دیگر نمودارهای UML
۱۶۶	۱-۷-۸ نمودار کلی تعامل (interaction overview diagram)
۱۶۹	تست‌های فصل هشتم
۱۷۳	پاسخ تست‌های فصل هشتم
۱۷۷	خودآزمایی

فصل نهم

۱۷۹	۱-۹ مقدمه
۱۷۹	۲-۹ تعیین صحت و تصدیق نرم‌افزار (Verification and Validation)
۱۸۰	۳-۹ مفاهیم و نکات اولیه
۱۸۲	۴-۹ آزمون جعبه‌ی سفید (white box testing)
۱۸۵	۱-۴-۹ آزمون مسیر پایه
۱۸۶	۲-۴-۹ آزمون حلقه (loop testing)
۱۸۷	۵-۹ آزمون جعبه سیاه (black box testing)
۱۸۹	۱-۵-۹ آزمون تجزیه‌ی هم‌ارزی (equivalence partitioning)
۱۹۰	۲-۵-۹ تحلیل مقادیر مرزی (Boundary Value Analysis)
۱۹۱	۶-۹ آزمون‌های خاص
۱۹۱	۱-۶-۹ آزمون سیستم‌های مبتنی بر معماری خادم / مخدوم

۱۹۲	۲-۶-۹ آزمون سیستم‌های بلادرنگ (real time systems)
۱۹۲	۳-۶-۹ آزمون واسط‌های گرافیکی کاربر (testing GUIs)
۱۹۲	۴-۶-۹ آزمون مستندات راهنما (testing help documentation)
۱۹۵	تست‌های فصل نهم
۱۹۷	پاسخ تست‌های فصل نهم
۱۹۹	خودآزمایی

فصل دهم

۲۰۱	۱-۱۰ مقدمه
۲۰۳	۲-۱۰ آزمون واحد (unit testing)
۲۰۴	۳-۱۰ آزمون یکپارچگی (integration testing)
۲۰۴	۱-۳-۱۰ آزمون غیرافزایشی (nonincremental testing)
۲۰۴	۲-۳-۱۰ آزمون افزایشی (incremental testing)
۲۰۶	۳-۳-۱۰ آزمون دود (smoke testing)
۲۰۷	۴-۳-۱۰ آزمون رگرسیون (regression testing)
۲۰۸	۵-۳-۱۰ آزمون اعتبارسنجی (validation test)
۲۰۸	۴-۱۰ آزمون سیستم (System testing)
۲۱۰	۵-۱۰ اشکال‌زدایی (debugging)
۲۱۳	تست‌های فصل دهم
۲۱۵	پاسخ تست‌های فصل دهم
۲۱۷	خودآزمایی

فصل یازدهم

۲۱۹	۱-۱۱ مقدمه
۲۱۹	۲-۱۱ حوزه‌های مختلف مدیریت پروژه
۲۲۰	۱-۲-۱۱ افراد (people)
۲۲۰	۲-۲-۱۱ محصول (product)
۲۲۱	۳-۲-۱۱ فرآیند (Process)
۲۲۱	۴-۲-۱۱ پروژه (project)
۲۲۲	۳-۱۱ انواع تیم‌های نرم‌افزاری
۲۲۳	۱-۳-۱۱ دیدگاه کنستانتین
۲۲۳	۲-۳-۱۱ دیدگاه مانتی (Mantie)
۲۲۵	۳-۳-۱۱ نکات تکمیلی
۲۲۶	۴-۱۱ اندازه‌ها و متریک‌ها در نرم‌افزار

۲۲۷	۱-۴-۱۱ متریک های فرآیند
۲۲۷	۲-۴-۱۱ متریک های پروژه
۲۲۸	۳-۴-۱۱ اندازه گیری در نرم افزار
۲۳۲	۴-۴-۱۱ متریک های کیفیت نرم افزار
۲۳۳	۵-۴-۱۱ کارایی رفع نقص (defect removal efficiency)
۲۳۴	۶-۴-۱۱ اندازه گیری نرم افزار بر مبنای هدف
۲۳۷	تست های فصل یازدهم
۲۴۱	پاسخ تست های فصل یازدهم
۲۴۵	خودآزمایی

فصل دوازدهم

۲۴۷	۱-۱۲ مقدمه
۲۴۸	۲-۱۲ تخمین پروژه
۲۴۸	۱-۲-۱۲ تعیین دامنه نرم افزار
۲۵۱	۲-۲-۱۲ تخمین اندازه نرم افزار
۲۵۳	۳-۲-۱۲ تخمین نیرو و هزینه
۲۵۶	۳-۱۲ زمان بندی پروژه
۲۵۷	۱-۳-۱۲ مراحل زمان بندی پروژه
۲۶۲	۲-۳-۱۲ پیگیری برنامه زمان بندی
۲۶۵	تست های فصل دوازدهم
۲۶۷	پاسخ تست های فصل دوازدهم
۲۶۹	خودآزمایی

فصل سیزدهم

۲۷۱	۱-۱۳ مقدمه
۲۷۲	۲-۱۳ مدیریت ریسک
۲۷۳	۱-۲-۱۳ انواع خطرات
۲۷۴	۲-۲-۱۳ شناسایی ریسک
۲۷۴	۳-۲-۱۳ تعیین تأثیر ریسک
۲۷۵	۴-۲-۱۳ تخفیف، نظارت و مدیریت بر ریسک (RMMM)
۲۷۶	۳-۱۳ تضمین کیفیت نرم افزار (Software Quality Assurance(SQA))
۲۷۷	۱-۳-۱۳ مفاهیم کیفیت
۲۷۸	۲-۳-۱۳ بازیابی های فنی رسمی (Formal Technical Review)
۲۸۰	۳-۳-۱۳ تیم SQA

۲۸۱	۴-۳-۱۳ قابلیت اطمینان نرم افزار (Software reliability)
۲۸۲	۴-۱۳ مدیریت پیکربندی نرم افزار (software configuration management)
۲۸۳	۱-۴-۱۳ خط مبنا (Base line)
۲۸۴	۲-۴-۱۳ کنترل نسخه (Version control)
۲۸۷	تست‌های فصل سیزدهم
۲۸۹	پاسخ تست‌های فصل سیزدهم
۲۹۱	خودآزمایی
	فصل چهاردهم
۲۹۳	۱-۱۴ مقدمه
۲۹۳	۲-۱۴ Rational unified process (RUP)
۲۹۴	۱-۲-۱۴ دیدگاه پویا (dynamic perspective)
۲۹۵	۲-۲-۱۴ دیدگاه ایستا (static perspective)
۲۹۷	۳-۲-۱۴ ارزیابی متدولوژی RUP
۲۹۷	۳-۱۴ متدولوژی SSADM
۲۹۸	۱-۳-۱۴ مطالعه امکان‌سنجی (feasibility study)
۲۹۸	۲-۳-۱۴ تحلیل نیازها (requirement analysis)
۲۹۹	۳-۳-۱۴ مشخصه‌ی نیازها (requirement specification)
۲۹۹	۴-۳-۱۴ مشخصه‌ی سیستم منطقی (Logical system specification)
۳۰۰	۵-۳-۱۴ طراحی فیزیکی
۳۰۰	۶-۳-۱۴ ارزیابی متدولوژی SSADM
۳۰۱	۴-۱۴ Scrum
۳۰۱	۱-۴-۱۴ ارزیابی متدولوژی Scrum
۳۰۳	۵-۱۴ متدولوژی XP
۳۰۳	۱-۵-۱۴ برنامه‌ریزی (planning)
۳۰۵	۲-۵-۱۴ طراحی (design)
۳۰۵	۳-۵-۱۴ کدنویسی (coding)
۳۰۶	۴-۵-۱۴ آزمون (testing)
۳۰۶	۵-۵-۱۴ ارزش‌های XP (XP values)
۳۰۷	۶-۵-۱۴ ارزیابی متدولوژی XP
۳۰۹	تست‌های فصل چهاردهم
۳۱۳	پاسخ تست‌های فصل چهاردهم
۳۱۷	خودآزمایی

۱-۱ مقدمه

با ظهور اولین کامپیوترها، نرم افزار نیز به عنوان جزئی جدانشدنی و حیاتی برای ارتباط و کار با سیستم های کامپیوتری مطرح شد. در طول گذشت پنجاه و چند سال، نقش نرم افزار، تغییرات چشمگیری داشته است. پیشرفت های گسترده در عملکرد سخت افزار، تغییرات شگرف در ساختار کامپیوترها و محاسبات، افزایش شدید حافظه های اصلی و جانبی سیستم و ...، همگی باعث پیچیدگی سیستم های مبتنی بر کامپیوتر شده اند. این پیچیدگی و البته تکامل باعث ظهور نتایج خیره کننده از موفقیت نرم افزار در حوزه های مختلف شده است؛ اما برای کسانی که این سیستم ها را تولید می کنند نیز مشکلات عدیده ای را نیز ایجاد نموده، به طوری که تولید محصولات نرم افزاری تبدیل به یکی از سنگین ترین پروژه های تولید محصول شده است. در طول این فصل ابتدا چند تعریف اولیه و مهم از نرم افزار و مهندسی آن ارائه می شود. سپس نرم افزار به عنوان یک محصول خاص و ویژه بررسی شده و به برخی از تفاوت های آن با دیگر ساخته های بشر اشاره می شود. انتهای این فصل نگاهی گذرا به حوزه ها و کاربردهای مختلف نرم افزار داشته و در نهایت برخی از باورهای قدیمی را به چالش کشیده و آن ها را اصلاح می نماید.

نرم افزار محصولی است که توسط مهندسین نرم افزار طراحی و تولید می شود. از دیدگاه فنی نرم افزار را می توان ترکیبی از برنامه، مستندات و داده دانست؛ اما از دید کاربر، نرم افزار محصولی است که با پردازش داده (اطلاعات خام)، اطلاعاتی را فراهم می آورد که جهان کاربر را بهتر می سازد. نرم افزار یا خود یک محصول است و یا گاهی وسیله ای برای ساخت یک محصول نرم افزاری می باشد. کامپایلرها و سیستم عامل ها از جمله نرم افزارهایی هستند که خود به عنوان یک وسیله برای تولید یک محصول نرم افزاری دیگر، مورد استفاده قرار می گیرند. همچنین در رابطه با مهندسی نرم افزار می توان گفت فناوری است که دربرگیرنده ی یک فرآیند، مجموعه ای از روش ها و یک مجموعه از ابزارها جهت توسعه محصول نرم افزاری است.

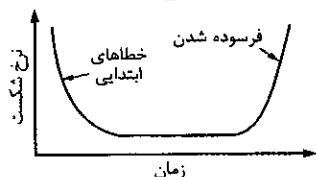
۱-۲ نرم افزار، محصولی متفاوت

نرم افزار بیش تر یک محصول منطقی است تا یک محصول فیزیکی. اکثر محصولات بشری عناصر فیزیکی بوده که به سادگی قابل لمس اند و هزینه ی تولید آن ها در تهیه مواد اولیه (material) آن ها نهفته است. در مقابل نرم افزار ماهیتی منطقی داشته و هزینه ی تولید آن در فرآیند مهندسی و طراحی آن نهفته است و نه در مواد اولیه. همین تفاوت اساسی، باعث بروز تفاوت های دیگری نیز می شود که در زیر بررسی شده اند.

- **نرم افزار توسعه (develop) می یابد یا طراحی (engineer) می شود؛ اما ساخته (manufacture) نمی شود.** در حقیقت توسعه ی نرم افزار متفاوت از ساخت یک محصول فیزیکی است. مثلاً کیفیت در ساخت یک محصول (سخت افزار)، بسیار وابسته به مواد اولیه آن محصول است، در حالی که کیفیت نرم افزار مستقیماً به کیفیت مراحل طراحی و مهندسی نگاشت

می‌شود. همان‌طور که گفته شد هزینه‌های نرم‌افزار در مهندسی آن متمرکز است، در حالی که هزینه‌ی ساخت یک محصول فیزیکی در تهیه مواد اولیه آن است. همین تفاوت‌ها باعث می‌شود تا مدیریت پروژه‌های نرم‌افزاری روندی متفاوت از مدیریت پروژه‌های ساخت محصول داشته باشد.

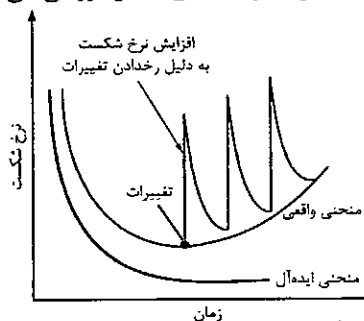
• نرم‌افزار مستهلک و فرسوده (wear out) نمی‌شود ولی با گذشت زمان کیفیت خود را از دست می‌دهد.



شکل ۱-۱ نمودار نرخ شکست در دوره‌ی حیات سخت‌افزار

پس از آن که یک محصول (سخت‌افزار) ساخته شد، در ابتدای دوره‌ی حیات خود شاهد خطاهای بسیاری است که باعث می‌شوند نرخ شکست محصول، در ابتدا بسیار بالا باشد. این شکست‌ها عموماً به دلیل خطاهایی است که در مرحله ساخت و یا طراحی وجود دارد. پس از گذشت زمان، این خطاها به سرعت برطرف شده و محصول به یک وضعیت پایدار و کم خطا می‌رسد. اما نرخ شکست سخت‌افزار بعد از سپری یک دوره‌ی زمانی دوباره بالا می‌رود؛ چرا که با گذشت زمان اجزای آن فرسوده شده و در نهایت سخت‌افزار مستهلک و بدون استفاده خواهد شد. شکل ۱-۱ نرخ شکست را در دوره‌ی حیات سخت‌افزار نشان می‌دهد.

اما وضعیت نرم‌افزار در دوره‌ی حیاتش چگونه است؟ می‌دانیم که نرم‌افزار فرسوده نمی‌شود، بنابراین انتظار داریم در بخش انتهایی نرخ شکست افزایش نیافت و نرم‌افزار همواره قابل استفاده باشد. در حالت ایده‌آل این امر درست است (شکل ۲-۱)، ولی متأسفانه بروز تغییرات عامل مهمی است که باعث می‌شود، نرم‌افزارها با گذشت زمان کیفیت خود را از دست داده و منسوخ شوند. نرم‌افزار همیشه شاهد تغییرات است و این تغییرات به دلایل مختلف حادث می‌شوند (در فصل بعد این دلایل بررسی می‌شوند).



شکل ۲-۱ نرخ شکست در دوره‌ی حیات نرم‌افزار

با بروز هر تغییر، نرم‌افزار از وضعیت پایدار خارج شده و نرخ شکست آن اوج می‌گیرد. با اصلاحات انجام شده توسط تیم نرم‌افزاری این وضعیت بهبود می‌یابد، اما با انجام تغییرات دیگر، باز هم شاهد اوج‌گیری نرخ شکست نرم‌افزار هستیم. متأسفانه تغییرات متناوب باعث می‌شود که نرم‌افزار از حالت ایده‌آل فاصله گرفته و منحنی نرخ شکست واقعی نرم‌افزار ایجاد شود (شکل ۲-۱).

• اگرچه صنعت به سمت مونتاژ اجزا (component)^۱ حرکت می‌کند، اما نرم‌افزار همچنان سفارشی ساخته می‌شود. استفاده از اجزا، مزایای زیادی را برای صنعت به ارمغان آورده است. کاهش زمان توسعه، کاهش هزینه‌ها و ارتقاء سطح اطمینان محصولات، از عمده‌ترین مزایای استفاده از اجزا می‌باشند. با این وجود متأسفانه صنعت مهندسی نرم‌افزار، نتوانسته به این تکنولوژی به شکل کامل دست یابد. هنوز هم بسیاری از بخش‌های یک نرم‌افزار، از ابتدا تحلیل و طراحی شده و امکان استفاده از اجزا (با قابلیت استفاده مجدد) در آن‌ها وجود ندارد. البته در سال‌های گذشته تلاش‌های زیادی در این زمینه انجام شده که حاصل آن‌ها شاخه‌ای از مدل‌های فرآیند، به نام مدل توسعه مبتنی بر اجزا^۲ است. امید است در ادامه‌ی راه نرم‌افزار، بتوان هر چه بیش‌تر از مزایای توسعه مبتنی بر اجزا بهره‌مند شد.

۳-۱ کاربردهای نرم‌افزار

امروزه هفت گروه اصلی برای کاربردهای نرم‌افزار شناخته می‌شود و نرم‌افزارها با توجه به محتوای اطلاعات، میزان دقت، صحت و قطعیتشان در یکی از این گروه‌ها طبقه‌بندی می‌شوند. در زیر نگاهی اجمالی به خصوصیات و ویژگی‌های هر یک از این گروه‌ها می‌اندازیم.

^۱. گاهی این واژه به مؤلفه نیز ترجمه می‌شود

^۲. Component based development

• نرم افزارهای سیستمی

نرم افزارهایی که با هدف ارائه سرویس به دیگر نرم افزارها توسعه یافته‌اند. این گروه از نرم افزارها به شکل چشمگیری با سخت افزار در تعاملند و عملیات همروند را با استفاده از تکنیک‌های زمان‌بندی، اشتراک‌گذاری منابع و مدیریت فرآیند، پشتیبانی می‌کنند.

اگر ترتیب و زمان دریافت ورودی، اجرای پردازش‌ها و تعیین خروجی‌ها مشخص باشد به آن نرم افزارهای سیستمی قطعی گوئیم (مانند کامپایلرها، ویرایشگرها و ابزارهای مدیریت فایل) و اگر ترتیب و زمان دریافت ورودی، اجرای پردازش‌ها و تعیین خروجی‌ها مشخص نباشد، به آن‌ها نرم افزارهای سیستمی غیر قطعی گفته می‌شود؛ مانند (اجزای سیستم عامل‌ها، درایورها و پردازشگرهای ارتباط از راه دور).

• نرم افزارهای کاربردی^۱

امروزه در هر تجارت شاهد نرم افزارهای کاربردی متنوعی هستیم که نیازهای آن تجارت را برطرف می‌کنند. نرم افزارهایی که اطلاعات مربوط به یک تجارت را مدیریت کرده و آن‌ها را پردازش می‌کنند و حتی گاه ابزاری جهت اتخاذ یک تصمیم تجاری می‌شوند. در بسیاری از این سیستم‌ها، نرم افزار با محیط و سخت افزارهای اطرافش در ارتباط بوده و داده‌ها و رویدادهای جهان واقعی را پردازش می‌کند. این پردازش‌ها باید خیلی سریع انجام شوند تا در اسرع وقت پاسخ را به محیط نرم افزار ارسال کنند. از این رو به این دسته از نرم افزارهای کاربردی، نرم افزارهای بلادرنگ^۲ گفته می‌شود. از جمله این نرم افزارها، نرم افزار پرتاب موشک در سیستم‌های نظامی، نرم افزار کنترل سیستم‌های فشار قوی برق در سیستم‌های صنعتی و نرم افزار تامین امنیت ساختمان (خانه‌ای امن^۳) در سیستم‌های تجاری هستند.

• نرم افزارهای مهندسی و علمی

این برنامه‌ها با پردازش‌های عددی سنگین و پیچیده شناسایی می‌شود. در حقیقت بخش اعظم پیچیدگی در الگوریتم و محاسبات برنامه متمرکز است تا در بخش‌های ورودی/خروجی. نرم افزارهای این حوزه در بسیاری از علوم از ستاره‌شناسی گرفته تا تحلیل فشار موتورهای صنعتی، نقش داشته و محاسبات پیچیده‌ای را انجام می‌دهند.

• نرم افزارهای جاسازی شده

این نرم افزارها کارکردهای محدود ولی بسیار سریعی دارند و در محصولات صنعتی که به بازار عرضه می‌شوند، گنجانده می‌شوند (مانند صفحه کلید کنترل در ماشین لباسشویی). کارکردهای این نرم افزارها بر اساس نیاز محصول و جهت کنترل خودکار محصول برنامه‌ریزی می‌شود.

• نرم افزار خط تولیدی^۴

این نرم افزارها بر طراحی و آماده‌سازی یک سرویس ویژه که مشتریان بسیاری خواهان آن هستند، تأکید می‌کند. این نرم افزارها بر رقابت‌های تجاری تمرکز دارند و تلاش می‌کنند محصولاتی تولید کنند که مشتری‌های زیادی در بازار داشته باشد.

• نرم افزارهای بر پایه‌ی وب

این نرم افزارها طیف وسیعی از برنامه‌هایی را شامل می‌شود که از طریق شبکه‌های کامپیوتری در دسترسند. در فرم ساده، این نرم افزارها می‌توانند شامل چند لینک بین صفحات و فایل‌های مختلف بوده و اطلاعاتی را به کاربر نمایش دهند. در فرم‌های پیچیده‌تر، این نرم افزارها می‌توانند محاسبات و پردازش‌هایی را در بستر شبکه انجام داده و یا حتی اطلاعات کاربران را در ساختار پایگاه داده مجتمع و یکپارچه کنند.

• نرم افزارهای هوش مصنوعی

این نرم افزارها از الگوریتم‌های غیر عددی برای حل مسائل پیچیده کمک می‌گیرند. مسائلی که با محاسبات متعارف قابل حل نبوده و روش حل آسانی ندارند. از حوزه‌های مختلف این نرم افزارها، می‌توان به نرم افزارهای ربات‌ها، شناسایی الگو (تصویر و صدا)، شبکه‌های عصبی مصنوعی و یا بازی‌های کامپیوتری اشاره کرد.

^۱. Application Software

^۲. Real time

^۳. Safe house

^۴. Product- Line Software

۴-۱ باورهای قدیمی ولی اشتباه!

در طی سالیان گذشته، باورهای بسیاری در بین افراد درگیر در علم نرم افزار انتشار یافته است. بسیاری از این باورها پایه و اساس صحیحی ندارند و تنها باعث گمراهی افراد پروژه می شوند. در اغلب آن ها مطلبی به ظاهر صحیح عنوان شده است، ولی در واقع هیچیک از آن ها صحیح نیست. بنابراین هر مهندس نرم افزار باید این باورها را بشناسد تا در طول اجرای پروژه به اشتباه از آن ها تبعیت نکند. در زیر به برخی از این باورها اشاره شده است.

۴-۱-۱ اشتباهات مدیران

• افراد من هنر توسعه ابزارهای نرم افزاری را دارند، ما برای آن ها جدیدترین کامپیوترها و سخت افزارها را می خریم.

واقعیت: ابزارهای مهندسی نرم افزار و نرم افزارهای کمکی (ابزارهای CASE) در رسیدن به کیفیت و بهره وری بالا بسیار مهم تر از سخت افزار هستند و سخت افزار در توسعه یک نرم افزار با کیفیت نقش چشمگیری ندارد.

• اگر از جدول زمان بندی عقب بمانیم، می توانیم به برنامه نویسان اضافه نموده و به جدول برسیم.

واقعیت: ارائه نرم افزار یک فرآیند مکانیکی تولید و ساخت نیست. اضافه کردن برنامه نویسان جدید کار را به تعویق می اندازد زیرا باید افراد قدیمی نکات پروژه را به افراد جدید آموزش دهند. عموماً در مواقعی که از زمان بندی پروژه عقب می مانیم بعضی از سرویس های نرم افزار را که ضرورت کمتری دارند از نسخه ی فعلی حذف کرده و آن ها را در نسخه های بعدی ارائه می دهیم.

• ما دارای کتابی هستیم که پر از استانداردها و مراحل کار تدوین و تکمیل یک نرم افزار است. اما آیا این کتاب همه آن چیزی را که افراد لازم دارند که بدانند را مهیا می کند؟

واقعیت: کتاب های مهندسی نرم افزار کامل بوده و مراحل تدوین نرم افزار را به روشنی بیان می کنند. سؤال اصلی اینجاست: آیا شرکت های نرم افزاری به درستی و دقت مراحل بیان شده در کتب نرم افزاری را بکار می بندند؟ آیا آن ها از فعالیت های جدید که در کتاب های نرم افزاری آمده است، استفاده می کنند؟

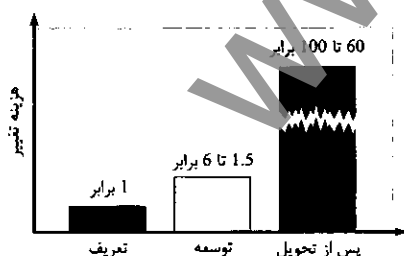
۴-۱-۲ اشتباهات مشتری

• وضعیت کلی اهداف مورد نظر پروژه برای شروع برنامه نویسی کافی است، می توانیم جزئیات را بعداً درست کنیم.

واقعیت: این دیدگاه دلیل اصلی شکست کارهای نرم افزاری است. جزئیات نیازها باید برای برنامه نویس و مهندسین نرم افزار در همان ابتدا روشن شود.

• موارد مورد نیاز و شرایط پروژه دائماً تغییر می کند، اما با تغییر می توان به راحتی سازگار شد؛ زیرا نرم افزار خاصیت تغییرپذیری دارد.

واقعیت: با گذشت زمان تغییرات اتفاق می افتند. بسته به این که در کدام مرحله از تولید نرم افزار این تغییرات حادث شوند هزینه های تحمیلی نرم افزار متفاوت خواهد بود. مثلاً اگر در مرحله تعریف، هزینه تحمیلی را برابر ۱ در نظر بگیریم، در مرحله توسعه، ۱.۵ تا ۶ و در مرحله پشتیبانی و پس از ساخت از ۶۰ تا ۱۰۰ خواهد بود (شکل ۳-۱). لذا با پیشرفت مراحل توسعه نرم افزار، دیگر تغییرات به سادگی اعمال نخواهند شد.



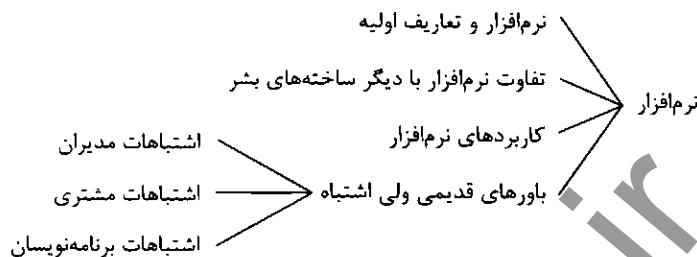
شکل ۳-۱ هزینه ی اعمال تغییرات در طول حیات نرم افزار

۴-۱-۳ اشتباهات برنامه نویسان

• وقتی برنامه نوشته و اجرا شد، دیگر کار تمام است.

واقعیت: اصل هزینه های نرم افزار در زمان پشتیبانی و پس از اولین تحویل (به مشتری) می باشد.

- تا وقتی که برنامه را اجرا نکنیم، هیچ راهی برای دسترسی به کیفیت آن وجود ندارد.
- واقعیت: بازبینیهای فنی رسمی^۱ (FTR) از زمان شروع پروژه تا زمان تحویل جهت تضمین کیفیت نرم افزار به کار می روند. در فصل آخر نکات بیش تری را در مورد FTR می آموزیم.
- تنها جزء قابل ارائه برای یک پروژه موفق، برنامه کاربردی است.
- واقعیت: مستندات دیگر جزء مهم است که هم در نگهداری نرم افزار مؤثر بوده و هم به عنوان راهنمای کاربر به کار می رود.



شکل ۴-۱ خلاصه ی فصل

^۱. Formal Technical Review