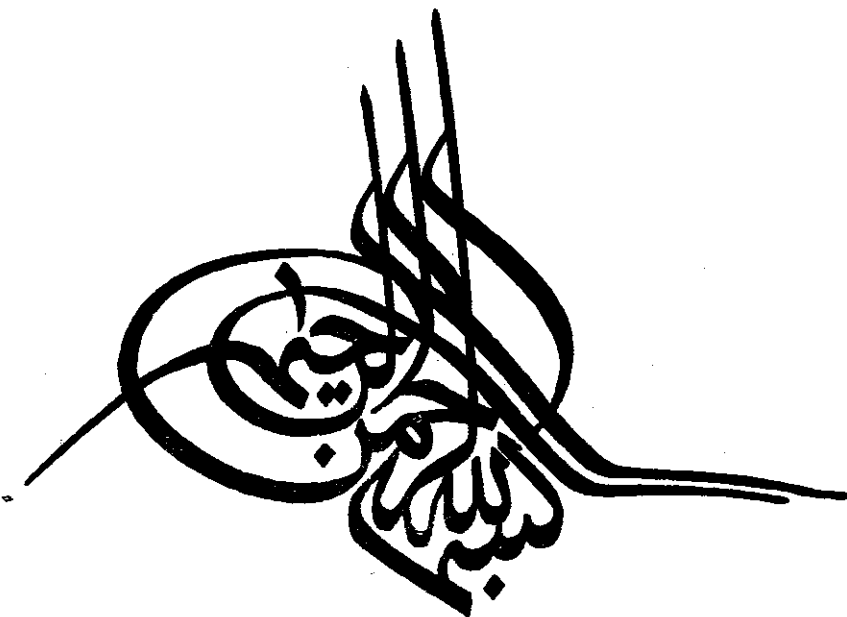


این کتاب تنها به خاطر حل مشکل دانشجویان پیام نور تبدیل به پی دی اف شد. همین جا از ناشر و نویسنده و تمام کسانی که با افزایش قیمت کتاب ما را مجبور به این کار کردند و یا متحمل ضرر شدند عذرخواهی می کنم.
گروهی از دانشجویان مهندسی کامپیوتر مرکز تهران



همدسی نرم افزار

رہیافتی برای یک اہل فن

جلد اول

پروفور راجہ اس. پرسمن
اثر

برکردان
نوید ہاشمی

تہ - بحار ۸۲

پرسمن، راجر، ۱۹۴۷-م.
مهندسی نرم افزار: رهیافتی برای یک اهل فن / اثر راجر اس. پرسمن:
برگردان نوید هاشمی طبّا.
تهران: گسترش علوم پایه، ۱۳۸۲.
ج ۲

ISBN 964-7817-38-x (دوره) - ISBN
964-7817-39-8 (ج. ۱) - ISBN 964-7817-40-1 (ج. ۲)
فهرست نویسی بر اساس اطلاعات فیپا.
عنوان اصلی:
Software Engineering: a
Practitioner's approach. 5th. ed, 2001.

این کتاب قبلاً تحت عنوان "مهندسی نرم افزار" با ترجمه عین اله
جعفرنژاد قمی و ابراهیم عامل محرابی توسط دانش نگار در سال
۱۳۸۱ چاپ شده است.
۱. نرم افزار -- مهندسی. الف. هاشمی طبّا، نوید، ۱۳۴۷ - مترجم. ب. عنوان.

۰۰۵/۱

۹۴ پ/۷۶/۷۵۸ QA
۱۳۸۲

۴۴۲۶۵۱-۸۱م

کتابخانه ملی ایران

نام کتاب :	مهندسی نرم افزار رهیافتی برای یک اهل فن (جلد اول)
مؤلف :	پروفسور راجر اس. پرسمن
مترجم:	نوید هاشمی طبّا
ناشر :	گسترش علوم پایه
طراح جلد :	حاتمی کیا
لیتوگرافی :	مهر
چاپ :	مهر
سال نشر :	۱۳۸۲
نوبت چاپ :	دوم
تیراژ :	۱۰۰۰ جلد
قیمت :	۱۸۰۰۰۰ ریال
کلمات کلیدی: نرم افزار، مهندسی نرم افزار، تحلیل طراحی، مکر، سیستم، مهندسی، مهندسی معکوس، مهندسی و مدیریت پروژه	
شابک: x-۳۸-۷۸۱۷-۹۶۴ (دوره دو جلدی)	ISBN: 964-7817-38-x (2 VOL-SET)
شابک: ۸-۳۹-۷۸۱۷-۹۶۴ (جلد اول)	ISBN: 964-7817-39-8 (VOL.2)

حق چاپ و نشر محفوظ و مخصوص ناشر می باشد.

تلفن انتشارات: ۱۵ ~ ۶۶۹۰۵۳۱۲ تلفکس: ۶۶۹۰۵۳۱۶-۲۱

دفتر سفارشات تهران: خ انقلاب، خ شهدای زاندارمری، بین فخر رازی و دانشگاه.

پلاک ۱۳۰، واحد ۱۵

مرکز پخش تهران: ۲۳ ~ ۶۶۹۷۵۸۳۹ تلفکس: ۶۶۹۷۷۰۴۷-۲۱

فروشگاه فجر تهران: خ انقلاب، بین فروردین و اردیبهشت، پ ۱۴۶۲ - ۶۶۴۰۴۹۸۳ (۰۲۱)

E-mail: info@Gostareh-op.com

www.Gostareh-op.com

بر سر در معبد علم نوشته اند:

هر آنکس که به درون گام می‌نهد

باید ایمان داشته باشد.

ماکس پلانک

فهرست بخش ها و فصل ها

شماره صفحه	عنوان
۱	مقدمه
۱۳	بخش اول محصول و فرآیند
۱۵	فصل ۱ محصول
۱۵	نگاه اجمالی
۱۷	۱-۱ نقش تکاملی نرم افزار
۱۹	۲-۱ نرم افزار
۱۹	۱-۲-۱ خصوصیات نرم افزار
۲۳	۲-۲-۱ کاربردهای نرم افزار
۲۵	۳-۱ نرم افزار : بحرانی در افق؟
۲۷	۴-۱ افسانه های نرم افزار
۳۰	۵-۱ خلاصه
۳۱	- مسایل و نکاتی برای تفکر و تعمق بیشتر
۳۲	فهرست منابع و مراجع
۳۲	خواندنیهای دیگر و منابع اطلاعاتی
۳۵	فصل ۲ فرآیند
۳۵	نگاه اجمالی
۳۶	۱-۲ مهندسی نرم افزار : یک فن آوری لایه ای
۳۷	۱-۱-۲ فرآیند، شیوه ها و ابزارها
۳۸	۲-۱-۲ یک دید کلی از مهندسی نرم افزار
۴۱	۲-۲ فرآیند نرم افزار
۴۶	۳-۲ مدلهای فرآیند نرم افزار
۴۸	۴-۲ مدل ترتیبی خطی
۵۱	۵-۲ مدل نمونه سازی
۵۳	۶-۲ مدل ساخت سریع برنامه ها (RAD)
۵۶	۷-۲ مدل های تکاملی فرآیند نرم افزار
۵۶	۱-۷-۲ مدل فزاینده (افزایشی)
۵۷	۲-۷-۲ مدل پیچشی (حلزونی)
۶۰	۳-۷-۲ مدل حلزونی WINWIN
۶۲	۴-۷-۲ مدل توسعه همروند

۶۵	۸-۲	توسعه مبتنی بر اجزاء
۶۶	۹-۲	مدل شیوه های رسمی
۶۷	۱۰-۲	فنون نسل چهارم
۶۹	۱۱-۲	فن آوری فرآیند
۷۰	۱۲-۲	محصول و فرآیند
۷۱	۱۳-۲	خلاصه
۷۲		مسایل و نکاتی برای تفکر و تعمق بیشتر
۷۳		فهرست منابع و مراجع
۷۵		خواندنیهای دیگر و منابع اطلاعاتی
۷۷		بخش دوم مدیریت پروژه های نرم افزاری
۷۹		فصل ۳ مفاهیم مدیریت پروژه
۷۹		نگاه اجمالی
۸۱	۱-۳	طیف مدیریت
۸۱	۱-۱-۳	افراد
۸۲	۲-۱-۳	محصول
۸۲	۳-۱-۳	فرآیند
۸۲	۴-۱-۳	پروژه
۸۳	۲-۳	افراد
۸۴	۱-۲-۳	بازیگران
۸۴	۲-۲-۳	رهبران تیم
۸۵	۳-۲-۳	تیم نرم افزار
۹۲	۴-۲-۳	مسائل هماهنگی و ارتباطات
۹۴	۳-۳	محصول
۹۵	۱-۳-۳	دامنه نرم افزار
۹۵	۲-۳-۳	تجزیه و شکستن مسئله
۹۶	۴-۳	فرآیند
۹۷	۱-۴-۳	تلفیق محصول و فرآیند
۹۹	۲-۴-۳	شکستن فرآیند
۱۰۰	۵-۳	پروژه
۱۰۲	۶-۳	اصل WSHH
۱۰۳	۷-۳	اقدامات بحرانی
۱۰۵	۸-۳	خلاصه
۱۰۶		مسایل و نکاتی برای تفکر و تعمق بیشتر
۱۰۸		فهرست منابع و مراجع

۱۰۸	خواندنیهای دیگر و منابع اطلاعاتی
	فصل ۴ متریک های پروژه فرآیند نرم افزار
۱۱۱	نگاه اجمالی
۱۱۱	۱-۴ اندازه ها، متریک ها و معیارها
۱۱۴	۲-۴ متریک ها در حوزه پروژه و فرآیند
۱۱۵	۱-۲-۴ متریک های فرآیند و بهبود فرآیند نرم افزار
۱۱۵	۲-۲-۴ متریک های پروژه
۱۲۱	۳-۴ اندازه گیری نرم افزار
۱۲۲	۱-۳-۴ متریک های مبتنی بر اندازه
۱۲۳	۲-۳-۴ متریک های مبتنی بر کارکرد (کارکرد-محور)
۱۲۵	۳-۳-۴ متریک های توسعه یافته امتیاز کارکردی
۱۲۷	۴-۴ تطبیق رهیافتهای مختلف متریک
۱۳۰	۵-۴ متریک های کیفیت نرم افزار
۱۳۲	۱-۵-۴ نگاهی اجمالی به فاکتورهای موثر بر کیفیت
۱۳۳	۲-۵-۴ اندازه گیری کیفیت
۱۳۴	۳-۵-۴ کارآیی رفع نقص
۱۳۵	۶-۴ متریک های انسجام و جامعیت در فرآیند مهندسی نرم افزار
۱۳۶	۱-۶-۴ استدلالی بر متریک های نرم افزاری
۱۳۷	۲-۶-۴ استقرار خط مبنا
۱۳۸	۳-۶-۴ جمع آوری، محاسبه و ارزیابی متریک ها
۱۳۸	۷-۴ توسعه و تکمیل واحدهای متریک و GQM
۱۳۹	۸-۴ مدیریت تغییرات : کنترل کیفیت آماری
۱۴۲	۹-۴ متریک های سازمان های کوچک
۱۴۴	۱۰-۴ برقراری یک برنامه برای متریک های نرم افزاری
۱۴۶	۱۱-۴ خلاصه
۱۴۸	مسایل و نکاتی برای تفکر و تعمق بیشتر
۱۴۹	فهرست منابع و مراجع
۱۵۱	خواندنیهای دیگر و منابع اطلاعاتی
۱۵۲	
	فصل ۵ طرح ریزی پروژه نرم افزاری
۱۵۳	نگاه اجمالی
۱۵۳	۱-۵ مشاهده برآوردها
۱۵۴	۲-۵ اهداف اصلی طرح ریزی پروژه
۱۵۶	۳-۵ دامنه نرم افزار
۱۵۶	

ت	فهرست عناوین
۱۵۷	۴-۵ دستیابی به اطلاعات مورد نیاز برای دامنه
۱۵۸	۱-۴-۵ امکان سنجی
۱۶۰	۲-۴-۵ یک مثال از تعیین دامنه
۱۶۲	۵-۵ منابع
۱۶۲	۱-۵-۵ منابع انسانی
۱۶۳	۲-۵-۵ منابع نرم افزاری با قابلیت استفاده مجدد
۱۶۵	۳-۵-۵ منابع محیطی
۱۶۵	۶-۵ برآورد پروژه های نرم افزاری
۱۶۷	۷-۵ فنون تجزیه
۱۶۷	۱-۷-۵ اندازه زدن نرم افزار
۱۶۹	۲-۷-۵ برآورد مبتنی بر مسئله
۱۷۱	۳-۷-۵ مثالی از برآورد مبتنی بر تعداد خطوط برنامه (LOC)
۱۷۳	۴-۷-۵ مثالی از برآورد مبتنی بر امتیاز کارکردی (FP)
۱۷۴	۵-۷-۵ برآورد مبتنی بر فرآیند
۱۷۵	۶-۷-۵ مثالی از برآورد مبتنی بر فرآیند
۱۷۶	۸-۵ مدل های برآورد تجربی
۱۷۷	۱-۸-۵ ساختار مدل های برآورد
۱۷۸	۲-۸-۵ مدل کوکومو (COCOMO)
۱۸۰	۳-۸-۵ معادله نرم افزار (فرمول ها)
۱۸۲	۹-۵ تصمیم گیری ساخت / خرید
۱۸۳	۱-۹-۵ ایجاد یک درخت تصمیم گیری
۱۸۴	۲-۹-۵ استفاده از منابع خارجی
۱۸۵	۱۰-۵ ابزارهای خودکار برآورد
۱۸۶	۱۱-۵ خلاصه
۱۸۷	مسایل و نکاتی برای تفکر و تعمق بیشتر
۱۸۹	فهرست منابع و مراجع
۱۸۹	خواندنیهای دیگر و منابع اطلاعاتی
۱۹۱	فصل ۶ مدیریت و تحلیل ریسک
۱۹۱	نگاه اجمالی
	۱-۶ راهبردهای واکنش بر ریسک در برابر راهبردهای
۱۹۳	پیش کنش در ریسک
۱۹۴	۲-۶ ریسک های نرم افزار
۱۹۵	۳-۶ شناسایی ریسک

ج	فهرست عناوین
۱۹۷	۱-۳-۶ ارزیابی ریسک کلی پروژه
۱۹۸	۲-۳-۶ اجزاء و محرک‌های ریسک
۲۰۰	۴-۶ برآورد ریسک
۲۰۰	۱-۴-۶ ساخت و توسعه یک جدول ریسک
۲۰۳	۲-۴-۶ ارزیابی میزان اثر ریسک
۲۰۴	۳-۴-۶ ارزیابی ریسک
۲۰۶	۵-۶ پالایش ریسک
۲۰۷	۶-۶ تخفیف، نظارت و مدیریت ریسک
۲۰۹	۷-۶ مخاطرات و ریسک‌های ایمنی
۲۱۰	۸-۶ طرح RMMM (تخفیف، نظارت و مدیریت ریسک)
۲۱۱	۹-۶ خلاصه
۲۱۳	مسایل و نکاتی برای تفکر و تعمق بیشتر
۲۱۵	فهرست منابع و مراجع
۲۱۵	خواندنیهای دیگر و منابع اطلاعاتی
۲۱۷	فصل ۷ زمان بندی و ردگیری پروژه
۲۱۸	نگاه اجمالی
۲۱۹	۱-۷ مفاهیم اولیه
۲۱۹	۱-۱-۷ توضیحی بر " دیرکرد "
۲۲۱	۲-۱-۷ اصول پایه
۲۲۳	۲-۷ رابطه میان افراد و نیروی کار
۲۲۴	۱-۲-۷ یک مثال
۲۲۵	۲-۲-۷ یک رابطه تجربی
۲۲۶	۳-۲-۷ توزیع نیروی کار
۲۲۶	۳-۷ یک مجموعه وظائف برای پروژه نرم افزاری
۲۲۸	۱-۳-۷ میزان سختی دقت
۲۲۹	۲-۳-۷ تعریف معیار تطابق
۲۳۰	۳-۳-۷ محاسبه یک معیار انتخاب مجموعه وظائف
	۴-۳-۷ تفسیر مقدار انتخاب مجموعه وظائف (TSS) و
۲۳۱	انتخاب مجموعه وظائف
۲۳۲	۴-۷ انتخاب وظائف مهندسی نرم افزار
۲۳۴	۵-۷ پالایش وظائف اصلی
۲۳۷	۶-۷ تعریف یک شبکه وظائف
۲۳۸	۷-۷ زمان بندی
۲۳۹	۱-۷-۷ نمودارهای خطی زمانی

ع	فهرست عناوین
۲۴۲	۲-۷-۷ ردگیری برنامه زمان بندی
۲۴۴	۸-۷ تحلیل مقدار بدست آمده (ارزش سنجی اکتسابی)
۲۴۶	۹-۷ ردگیری خطا
۲۴۸	۱۰-۷ طرح پروژه
۲۴۸	۱۱-۷ خلاصه
۲۴۹	مسایل و نکاتی برای تفکر و تعمق بیشتر
۲۵۲	فهرست منابع و مراجع
۲۵۲	خواندنیهای دیگر و منابع اطلاعاتی
۲۵۳	فصل ۸ تضمین کیفیت نرم افزار
۲۵۳	نگاه اجمالی
۲۵۵	۱-۸ مفاهیم کیفیت
۲۵۶	۱-۱-۸ کیفیت
۲۵۷	۲-۱-۸ کنترل کیفیت
۲۵۷	۳-۱-۸ تضمین کیفیت
۲۵۸	۴-۱-۸ هزینه کیفیت
۲۶۰	۲-۸ حرکت کیفی
۲۶۲	۳-۸ تضمین کیفیت نرم افزار
۲۶۱	۱-۳-۸ موضوعات زمینه
۲۶۳	۲-۳-۸ فعالیتهای تضمین کیفیت نرم افزار (SQA)
۲۶۴	۴-۸ بازیینی های نرم افزار
۲۶۶	۱-۴-۸ تاثیر عیوب بر هزینه نرم افزاری
۲۶۹	۲-۴-۸ تشدید نقص و برطرف کردن آن
۲۶۹	۵-۸ بازیینی های فنی رسمی
۲۷۰	۱-۵-۸ نشست بازیینی
۲۷۱	۲-۵-۸ گزارش بازیینی ها و ثبت موضوعات
۲۷۲	۳-۵-۸ رهنمودهای بازیینی
۲۷۴	۶-۸ رهیافت های رسمی برای تضمین کیفیت نرم افزار
۲۷۴	۷-۸ تضمین کیفیت آماری نرم افزار
۲۷۷	۸-۸ قابلیت اطمینان نرم افزار
۲۷۷	۱-۸-۸ اندازه گیری قابلیت اطمینان و قابلیت دسترسی
۲۷۹	۲-۸-۸ ایمنی نرم افزار
۲۸۱	۹-۸ مصونیت نرم افزار در برابر اشتباه (نرم افزار ضد خطا)
۲۸۴	۱۰-۸ استانداردهای کیفیت ایزو ۹۰۰۱
۲۸۴	۱-۱۰-۸ رهیافت ایزو جهت سیستم های تضمین کیفیت

۲۸۵	۸-۱۰-۲ استاندارد ایزو ۹۰۰۱
۲۸۶	۸-۱۱ طرح تضمین کیفیت نرم افزار (SQA)
۲۸۷	۸-۱۲ خلاصه
۲۸۸	مسایل و نکاتی برای تفکر و تعمق بیشتر
۲۹۰	فهرست منابع و مراجع
۲۹۱	خواندنیهای دیگر و منابع اطلاعاتی
۲۹۳	فصل ۹ مدیریت پیکربندی نرم افزار
۲۹۳	نگاه اجمالی
۲۹۵	۹-۱ مدیریت پیکربندی نرم افزار (SCM)
۲۹۶	۹-۱-۱ خطوط مبنا
۲۹۸	۹-۱-۲ اقلام پیکربندی نرم افزار
۲۹۹	۹-۲ فرآیند مدیریت پیکربندی نرم افزار
۳۰۰	۹-۳ شناسایی اشیاء در پیکربندی نرم افزار
۳۰۲	۹-۴ کنترل نسخه
۳۰۵	۹-۵ کنترل تغییرات
۳۰۹	۹-۶ واریسی پیکربندی
۳۱۰	۹-۷ گزارش وضعیت
۳۱۱	۹-۸ استانداردهای مدیریت پیکربندی نرم افزار
۳۱۱	۹-۹ خلاصه
۳۱۳	مسایل و نکاتی برای تفکر و تعمق بیشتر
۳۱۴	فهرست منابع و مراجع
۳۱۴	خواندنیهای دیگر و منابع اطلاعاتی
۳۱۷	بخش سوم روشهای متعارف مهندسی نرم افزار
۳۱۹	فصل ۱۰ مهندسی سیستم
۳۱۹	نگاه اجمالی
۳۲۱	۱۰-۱ سیستم های مبتنی بر کامپیوتر
۳۲۳	۱۰-۲ سلسله مراتب مهندسی سیستم
۳۲۴	۱۰-۲-۱ مدل سازی سیستم
۳۲۶	۱۰-۲-۲ شبیه سازی سیستم
۳۲۷	۱۰-۳ مهندسی فرآیند تجاری : یک دید کلی
۳۳۱	۱۰-۴ مهندسی محصول : یک دید کلی
۳۳۳	۱۰-۵ مهندسی نیازمندیها
۳۳۳	۱۰-۵-۱ تعیین نیازمندیها

ح	فهرست عناوین
۲۳۵	۲-۵-۱۰ تحلیل نیازمندیها و مذاکرات مربوطه
۲۳۷	۳-۵-۱۰ تعیین مشخصات نیازمندیها
۲۳۷	۴-۵-۱۰ مدل سازی سیستم
۲۳۸	۵-۵-۱۰ اعتبارسنجی نیازمندیها
۲۳۹	۶-۵-۱۰ مدیریت نیازمندیها
۲۴۱	۶-۱۰ مدل سازی سیستم
۲۴۵	۷-۱۰ خلاصه
۲۴۷	مسایل و نکاتی برای تفکر و تعمق بیشتر
۲۴۹	فهرست منابع و مراجع
۲۴۹	خواندنیهای دیگر و منابع اطلاعاتی
۲۵۱	فصل ۱۱ اصول و مفاهیم تحلیل
۲۵۱	نگاه اجمالی
۲۵۲	۱-۱۱ تحلیل نیازمندیها
۲۵۵	۲-۱۱ تعیین نیازمندیهای نرم افزار
۲۵۵	۱-۲-۱۱ راه اندازی فرآیند
۲۵۶	۲-۲-۱۱ فنون تسهیل مشخص سازی کاربردی
۲۶۰	۳-۲-۱۱ تنظیم کارکرد کیفیت
۲۶۲	۴-۲-۱۱ USE-CASE ها
۲۶۵	۳-۱۱ اصول تحلیل
۲۶۶	۱-۳-۱۱ میدان اطلاعات
۲۶۸	۲-۳-۱۱ مدل سازی
۲۷۰	۳-۳-۱۱ افراز (تجزیه)
۲۷۲	۴-۳-۱۱ دیدگاه های اساسی و پیاده سازی
۲۷۳	۴-۱۱ نمونه سازی نرم افزار
۲۷۴	۱-۴-۱۱ انتخاب رهیافت نمونه سازی
۲۷۵	۲-۴-۱۱ شیوه ها و ابزارهای نمونه سازی
۲۷۶	۵-۱۱ تعیین مشخصات
۲۷۶	۱-۵-۱۱ اصول تعیین مشخصات
۲۷۷	۲-۵-۱۱ بازنمایی
۲۷۸	۳-۵-۱۱ تعیین مشخصات نیازمندیهای نرم افزار
۲۷۹	۶-۱۱ بازبینی مشخصات

فهرست عناوین	خ
۳۸۰	۷-۱۱ خلاصه
۳۸۲	مسایل و نکاتی برای تفکر و تعمق بیشتر
۳۸۴	فهرست منابع و مراجع
۳۸۴	خواندنیهای دیگر و منابع اطلاعاتی
۳۸۷	فصل ۱۲ مدل سازی تحلیل.
۳۸۷	نگاه اجمالی
۳۸۹	۱-۱۲ تاریخچه ای مختصر
۳۹۰	۲-۱۲ عناصر مدل تحلیل
۳۹۲	۳-۱۲ مدل سازی داده ها
۳۹۲	۱-۳-۱۲ اشیاء داده ای، صفات خاصه و روابط
۳۹۷	۲-۳-۱۲ کاردینالیتیه و مدالیتیه
۴۰۰	۳-۳-۱۲ نمودارهای موجودیت / رابطه
۴۰۱	۴-۱۲ مدل سازی کارکردی و جریان اطلاعات
۴۰۳	۱-۴-۱۲ نمودارهای جریان داده ها
۴۰۵	۲-۴-۱۲ توسعه سیستمهای مبتنی بر زمان واقعی (Real - time)
۴۰۵	۳-۴-۱۲ فعالیتهای تکمیلی "وارد" و "ملور"
۴۰۸	۴-۴-۱۲ فعالیتهای تکمیلی "هتلی" و "پیربهای"
۴۱۱	۵-۱۲ مدل سازی رفتاری
۴۱۳	۶-۱۲ مکانیک تحلیل ساخت یافته
۴۱۴	۱-۶-۱۲ ایجاد یک نمودار موجودیت / رابطه
۴۱۸	۲-۶-۱۲ ایجاد یک مدل جریان داده ها
۴۲۰	۳-۶-۱۲ ایجاد یک مدل جریان کنترل
۴۲۱	۴-۶-۱۲ تعیین مشخصات کنترل
۴۲۳	۵-۶-۱۲ تعیین مشخصات فرآیند
۴۲۵	۷-۱۲ فرهنگ داده ها (واژه نامه داده ها)
۴۲۸	۸-۱۲ دیگر شیوه های تحلیل سنتی
۴۲۸	۹-۱۲ خلاصه
۴۳۰	مسایل و نکاتی برای تفکر و تعمق بیشتر
۴۳۲	فهرست منابع و مراجع
۴۳۲	خواندنیهای دیگر و منابع اطلاعاتی

۴۳۵	فصل ۱۳ اصول و مفاهیم طراحی
۴۳۵	نگاه اجمالی
۴۳۷	۱-۱۳ طراحی نرم افزار و مهندسی نرم افزار
۴۳۹	۲-۱۳ فرآیند طراحی
۴۳۹	۱-۲-۱۳ طراحی و کیفیت نرم افزار
۴۴۰	۲-۲-۱۳ سیر تکاملی طراحی نرم افزار
۴۴۱	۳-۱۳ اصول طراحی
۴۴۲	۴-۱۳ مفاهیم طراحی
۴۴۲	۱-۴-۱۳ تجرید
۴۴۴	۲-۴-۱۳ پالایش
۴۴۵	۳-۴-۱۳ پیمانه سازی
۴۴۸	۴-۴-۱۳ معماری نرم افزار
۴۴۹	۵-۴-۱۳ سلسله مراتب کنترل
۴۵۱	۶-۴-۱۳ تجزیه ساختاری
۴۵۲	۷-۴-۱۳ ساختار داده ها
۴۵۵	۸-۴-۱۳ رویه نرم افزار
۴۵۵	۹-۴-۱۳ پنهان سازی اطلاعات
۴۵۶	۵-۱۳ طراحی موثر پیمانه ای (ماجولار)
۴۵۶	۱-۵-۱۳ استقلال کارکردی
۴۵۷	۲-۵-۱۳ چسبندگی
۴۵۸	۳-۵-۱۳ پیوستگی
۴۶۰	۶-۱۳ ابداعات طراحی برای پیمانه سازی موثر و کارآ
۴۶۲	۷-۱۳ مدل طراحی
۴۶۲	۸-۱۳ مستندسازی طراحی
۴۶۴	۹-۱۳ خلاصه
۴۶۵	مسایل و نکاتی برای تفکر و تعمق بیشتر
۴۶۷	فهرست منابع و مراجع
۴۶۸	خواندنیهای دیگر و منابع اطلاعاتی
۴۷۱	فصل ۱۴ طراحی معماری
۴۷۱	نگاه اجمالی
۴۷۲	۱-۱۴ معماری نرم افزار
۴۷۲	۱-۱-۱۴ معماری چیست ؟

۴۷۴	چرا معماری از اهمیت برخوردار است؟	۲-۱-۱۴
۴۷۴	طراحی داده ها	۲-۱۴
۴۷۵	مدل سازی داده، ساختار داده، پایگاه داده و انبار داده ها	۱-۲-۱۴
۴۷۷	طراحی تفصیلی داده ها (در سطح اجزاء)	۲-۲-۱۴
۴۷۸	سبک های معماری	۳-۱۴
۴۷۹	گروه بندی مختصر سبک ها و الگوها	۱-۳-۱۴
۴۸۲	سازمان دهی و پالایش	۲-۳-۱۴
۴۸۲	تحلیل طراحی های انواع معماری های جایگزین	۴-۱۴
۴۸۲	یک شیوه تحلیل توازن معماری	۱-۴-۱۴
۴۸۵	رهنمود کمی برای طراحی معماری	۲-۴-۱۴
۴۸۷	پیچیدگی معماری	۳-۴-۱۴
۴۸۸	نگاشت نیازمندیها در یک معماری نرم افزار	۵-۱۴
۴۸۹	جریان تبدیلات	۱-۵-۱۴
۴۹۰	جریان تراکنش	۲-۵-۱۴
۴۹۰	نگاشت تبدیلات	۶-۱۴
۴۹۱	یک مثال	۱-۶-۱۴
۴۹۱	گامهای طراحی	۲-۶-۱۴
۵۰۰	نگاشت تراکنش ها	۷-۱۴
۵۰۱	یک مثال	۱-۷-۱۴
۵۰۱	گامهای طراحی	۲-۷-۱۴
۵۰۵	پالایش طراحی معماری	۸-۱۴
۵۰۶	خلاصه	۹-۱۴
۵۰۸	مسایل و نکاتی برای تفکر و تعمق بیشتر	
۵۱۰	فهرست منابع و مراجع	
۵۱۱	خواندنیهای دیگر و منابع اطلاعاتی	
۵۱۳	فصل ۱۵ طراحی رابط کاربر	
۵۱۳	نگاه اجمالی	
۵۱۴	قواعد طلایی	۱-۱۵
۵۱۵	اعطای کنترل به کاربر	۱-۱-۱۵
۵۱۶	کاستن از بار حافظه کاربر	۲-۱-۱۵
۵۱۷	سازگار سازی رابط	۳-۱-۱۵
۵۱۸	طراحی رابط کاربر	۲-۱۵

ر	فهرست عناوین
۵۱۸	۱-۲-۱۵ مدل های طراحی رابط
۵۲۰	۲-۲-۱۵ فرآیند طراحی رابط کاربر
۵۲۱	۳-۱۵ تحلیل وظائف و مدل سازی
۵۲۳	۴-۱۵ فعالیتهای طراحی رابط
۵۲۳	۱-۴-۱۵ تعریف اشیاء و اقدامات رابط
۵۲۶	۲-۴-۱۵ موضوعات طراحی
۵۲۹	۵-۱۵ ابزارهای پیاده سازی
۵۳۰	۶-۱۵ ارزیابی طراحی
۵۳۲	۷-۱۵ خلاصه
۵۳۴	مسایل و نکاتی برای تفکر و تعمق بیشتر
۵۳۶	فهرست منابع و مراجع
۵۳۶	خواندنیهای دیگر و منابع اطلاعاتی
۵۳۹	فصل ۱۶ طراحی تفصیلی (سطح اجزاء)
۵۳۹	نگاه اجمالی
۵۴۱	۱-۱۶ برنامه سازی ساختیافته
۵۴۲	۱-۱-۱۶ علائم طراحی گرافیکی
۵۴۵	۲-۱-۱۶ علائم طراحی جدولی
۵۴۷	۳-۱-۱۶ زبان طراحی برنامه (PDL)
۵۴۹	۴-۱-۱۶ مثالی از زبان طراحی برنامه
۵۵۱	۲-۱۶ مقایسه علائم طراحی
۵۵۳	۳-۱۶ خلاصه
۵۵۴	مسایل و نکاتی برای تفکر و تعمق بیشتر
۵۵۶	فهرست منابع و مراجع
۵۵۷	خواندنیهای دیگر و منابع اطلاعاتی

فهرست شکل ها و جداول

شماره

عنوان

صفحه

فصل ۱

- شکل ۱-۱ منحنی شکست سخت افزار ۲۰
شکل ۲-۱ منحنی ایده آل و واقعی شکست برای نرم افزار ۲۲
شکل ۳-۱ اثر تغییر ۲۸

فصل ۲

- شکل ۱-۲ لایه های مهندسی نرم افزار ۲۸
شکل ۲-۲ فرآیند نرم افزار ۴۲
شکل ۳-۲ الف) مراحل یک چرخه حل مسئله [RAC 95] ۴۷
ب) مراحل میانی چرخه حل مسئله [RAC 95] ۴۷
شکل ۴-۲ مدل ترتیبی خطی ۴۹
شکل ۵-۲ پارادایم ساخت نمونه ۵۱
شکل ۶-۲ مدل توسعه سریع برنامه ها (RAD) ۵۴
شکل ۷-۲ مدل افزایشی ۵۷
شکل ۸-۲ یک مدل حلزونی ۵۹
شکل ۹-۲ مدل حلزونی WINWIN ۶۱
شکل ۱۰-۲ یک عنصر از مدل فرآیند همروند ۶۳
شکل ۱۱-۲ توسعه مبتنی بر اجزاء ۶۵

فصل ۳

- شکل ۱-۲ ارزش کاربرد فزون ارتباط و هماهنگ سازی ۹۳
شکل ۲-۲ مرتبط ساختن مسئله و فرآیند ۹۸

فصل ۴

- شکل ۱-۴ تعیین کیفیت نرم افزار و مؤثر بودن سازمانی (منطبق بر [PAU94]) ۱۱۶
شکل ۲-۴ دلایل عیوب و منشاء آنها برای چهار پروژه نرم افزاری [GRA 94] ۱۱۸
شکل ۳-۴ یک نمودار استخوان ماهی (مطابق با [GRA 92]) ۱۲۰
شکل ۴-۴ متریک های مبتنی بر سایز ۱۲۴
شکل ۵-۴ محاسبه امتیاز کارکرد (عملیاتی) ۱۲۶
شکل ۶-۴ تعیین میزان پیچیدگی یک تبدیل برای از امتیازات عملیاتی سه بعدی [WHI95] ۱۲۹
شکل ۷-۴ فرآیند جمع آوری متریک های نرم افزاری ۱۴۰
شکل ۸-۴ داده های متریک برای خطاهای پوشش داده نشده در هر ساعت بازبینی ۱۴۰
شکل ۹-۴ چارت کنترل رنج حرکت ۱۴۱
شکل ۱۰-۴ چارت کنترل افراد ۱۴۱

فصل ۵

- ۱۶۱ شکل ۵-۱ یک سیستم مرتب سازی خط نقاله
- ۱۶۲ شکل ۵-۲ منابع پروژه
- ۱۷۲ شکل ۵-۳ جدول برآورد و تخمین برای روش تعداد خطوط بر نامه (LOC)
- ۱۷۴ شکل ۵-۴ مقادیر برآورد شده در حوزه اطلاعات
- ۱۷۶ شکل ۵-۵ جدول برآوردهای مبتنی بر فرآیند
- ۱۷۹ جدول ۵-۱ وزن پیچیدگی برای انواع اشیاء [BOE96]
- ۱۸۰ جدول ۵-۲ نرخ های بهره وری برای امتیازات اشیاء [BOE96]

فصل ۶

- ۱۹۹ شکل ۶-۱ ارزیابی تاثیرات [BOE89]
- ۲۰۱ شکل ۶-۲ مثالی از مخاطرات پیش از مرتب سازی
- ۲۰۲ شکل ۶-۳ ریسک و امور مدیریتی
- ۲۰۵ شکل ۶-۴ سطح معرف ریسک
- ۲۱۱ شکل ۶-۵ نمونه ای از صفحه اطلاعات دیسک (RIS) [WIL97]

فصل ۷

- ۲۲۲ شکل ۷-۱ وظایف توسعه مفهومی در یک مدل خطی ترتیبی
- ۲۲۵ شکل ۷-۲ وظایف توسعه مفهومی با استفاده از مدل تکاملی (افزایشی)
- ۲۲۷ شکل ۷-۳ یک شبکه کاری (وظیفه ای) برای توسعه مفهومی
- ۲۴۱ شکل ۷-۴ مثالی از نمودار خطی زمانی (گانت چارت)
- ۲۴۲ شکل ۷-۵ مثالی از جدول پروژه
- ۲۳۰ جدول ۷-۱ محاسبات انتخابگر مجموعه وظایف (TSS)
- ۲۳۱ جدول ۷-۲ مثالی از محاسبات انتخابگر مجموعه وظایف (TSS)

فصل ۸

- ۲۵۹ شکل ۸-۱ هزینه نسبی اصلاح یک خطا
- ۲۶۷ شکل ۸-۲ مدل تشدید و توسعه نقص
- ۲۶۸ شکل ۸-۳ تشدید نقص و عیب بدون بازایی
- ۲۶۸ شکل ۸-۴ تشدید نقص و عیب با بازایی

فصل ۹

- ۲۹۶ شکل ۹-۱ SCI های خط مبنا و پایگاه داده های پروژه
- ۲۹۸ شکل ۹-۲ اشیاء پیکربندی
- ۳۰۱ شکل ۹-۳ گراف تکامل
- ۳۰۴ شکل ۹-۴ مخزن شیء، باز نمای اشیاء، تنوعات و نسخه ها [REI89]
- ۳۰۸ شکل ۹-۵ فرآیند کنترل تغییر
- ۳۰۹ شکل ۹-۶ کنترل دسترسی و هماهنگی

فصل ۱۰

۳۲۲	شکل ۱-۱۰ سلسله مراتب مهندسی سیستم
۳۲۹	شکل ۲-۱۰ سلسله مراتب مهندسی فرآیند تجاری
۳۳۲	شکل ۳-۱۰ سلسله مراتب مهندسی محصول
۳۴۰	شکل ۴-۱۰ جدول ژنریک ردیابی (صحت)
۳۴۲	شکل ۵-۱۰ قالب مدل سیستم [HAT87]
۳۴۳	شکل ۶-۱۰ نمودار متن توسعه یافته برای Class (توسعه یافته)
۳۴۴	شکل ۷-۱۰ نمودار جریان سیستم برای class توسعه یافته (سیستم خط نقاله و مرتب سازی)
۳۴۶	شکل ۸-۱۰ ساخت یک سلسله مراتب SFD (نمودار جریان سیستم)

فصل ۱۱

۳۵۴	شکل ۱-۱۱ تحلیل به منزله پلی است بین مهندسی سیستم و طراحی نرم افزار
۳۶۳	شکل ۲-۱۱ پائل کنترل خانه امن
۳۶۸	شکل ۳-۱۱ جریان اطلاعات و تبدیلات
۳۷۰	شکل ۴-۱۱ پارتیشن بندی افقی کارکرد خانه امن
۳۷۲	شکل ۵-۱۱ پارتیشن بندی عمودی کارکرد خانه امن
۳۷۵	شکل ۶-۱۱ انتخاب رهیافت مناسب برای مدل نمونه

فصل ۱۲

۳۹۱	شکل ۱-۱۲ ساختار یک مدل تحلیلی
۳۹۳	شکل ۲-۱۲ اشیاء داده ای، صفت خاصه و رابطه ها
۳۹۵	شکل ۳-۱۲ بازنمایی اشیاء داده ای در قالب جدول
۳۹۶	شکل ۴-۱۲ رابطه ها
۳۹۶	شکل ۵-۱۲ کاردینالیت و مدالیت
۳۹۷	شکل ۶-۱۲ یک ERD (نمودار رابطه / موجودیت) ساده و جدول اشیاء داده ای (توجه شود که در این ERD رابطه "می سازد" یا یک لوزی مشخص شده)
۳۹۸	شکل ۷-۱۲ یک ERD (نمودار رابطه / موجودیت) توسعه یافته
۳۹۹	شکل ۸-۱۲ سلسله مراتب نوع اشیاء داده ای
۴۰۲	شکل ۹-۱۲ اشیاء داده ای شرکت پذیر
۴۰۳	شکل ۱۰-۱۲ مدل جریان اطلاعات
۴۰۶	شکل ۱۱-۱۲ پالایش جریان اطلاعات
۴۰۷	شکل ۱۲-۱۲ جریان داده و کنترل با استفاده از علائم وارد و ملور [WAR85]
۴۰۹	شکل ۱۴-۱۲ رابط میان مدلهای داده ای و کنترلی
۴۱۰	شکل ۱۵-۱۲ شرایط داده
۴۱۱	شکل ۱۶-۱۲ سطح یک CDF (نمودار جریان کنترل) برای نرم افزار نسخه برداری
۴۱۳	شکل ۱۷-۱۲ نمودار انتقال وضعیت برای نرم افزار نسخه برداری (فتوکیبی)
۴۱۵	شکل ۱۸-۱۲ برقراری ارتباطات
۴۱۶	شکل ۱۹-۱۲ ساخت و ارتباطات و کاردینالیت / مدالیت
۴۱۶	شکل ۲۰-۱۲ نمودار DFD سطح متن
۴۱۷	شکل ۲۱-۱۲ سطح DFD (نمودار جریان داده) برای خانه امن

- ۴۱۷ شکل ۱۲-۲۲ سطح دوم DFD (نمودار جریان داده ها) که فرآیند ناظر بر حس گرها را پالایش می کند
- ۴۲۲ شکل ۱۲-۲۳ سطح یک CFD (نمودار روند کنترل) خانه امن
- ۴۲۳ شکل ۱۲-۲۴ نمودار انتقال وضعیت برای خانه امن
- ۴۲۴ شکل ۱۲-۲۵ جدول فعال سازی فرآیند خانه امن

فصل ۱۳

- ۴۲۷ شکل ۱۳-۱ برگردان مدل تحلیلی به طراحی نرم افزار
- ۴۴۵ شکل ۱۳-۲ پیمانه شدن و هزینه نرم افزار
- ۴۴۸ شکل ۱۳-۳ شناخت اصطلاحات ساختار برای یک سبک معماری فراخوانی و بازگشت
- ۴۵۲ شکل ۱۳-۴ پارتیشن بندی ساختاری
- ۴۵۳ شکل ۱۳-۵ رویه لایه لایه شدن
- ۴۵۸ شکل ۱۳-۶ انواع اتصال و چسبندگی
- ۴۶۲ شکل ۱۳-۷ ساختارهای برنامه

فصل ۱۴

- ۴۸۰ شکل ۱۴-۱ معماری متمرکز داده ها
- ۴۸۱ شکل ۱۴-۲ معماری جریان داده ها
- ۴۸۲ شکل ۱۴-۳ معماری لایه لایه شده
- ۴۹۰ شکل ۱۴-۴ روند تراکنش
- ۴۹۱ شکل ۱۴-۵ سطح متن برای DFD (نمودار جریان داده ها) خانه امن
- ۴۹۲ شکل ۱۴-۶ DFD سطح ۱ برای خانه امن
- ۴۹۳ شکل ۱۴-۷ سطح دوم DFD (نمودار جریان داده ها) برای فرآیند حسگرهای نمایشگر
- ۴۹۴ شکل ۱۴-۸ سطح سوم DFD (نمودار جریان داده ها) برای خانه امن با مرزهای روند
- ۴۹۶ شکل ۱۴-۹ سطح نخست حس گرهای نمایشگر
- ۴۹۸ شکل ۱۴-۱۰ فاکتور بندی سطح دوم برای حس گرهای نظارتی
- ۵۰۰ شکل ۱۴-۱۱ "نمودار نخست" ساختار برنامه برای حس گرهای نمایش دهنده
- ۵۰۱ شکل ۱۴-۱۲ ساختار پالایش شده برنامه برای حس گرهای نمایش دهنده
- ۵۰۲ شکل ۱۴-۱۳ سطح ۲ برای زیر سیستم محاوره کاربر با مرزهای جریان
- ۵۰۳ شکل ۱۴-۱۴ نگاشت تراکنش
- ۵۰۴ شکل ۱۴-۱۵ عوامل سطح نخست برای زیر سیستم رابط کاربر
- ۵۰۵ شکل ۱۴-۱۶ معماری تکرار - اول برای زیر سیستم رابط کاربر

فصل ۱۵

- ۵۱۹ شکل ۱۵-۱ فرآیند طراحی رابط کاربر
- ۵۲۶ شکل ۱۵-۲ آرایش اولیه صفحه نمایش
- ۵۳۱ شکل ۱۵-۳ چرخه ارزیابی طراحی رابط

فصل ۱۶

۵۴۱	شکل ۱۶-۱ ساختمان های روند نما
۵۴۴	شکل ۱۶-۲ ساختمانهای تودر تو
۵۴۵	شکل ۱۶-۳ ساختمانهای نمودار جعبه ای
۵۴۶	شکل ۱۶-۴ ساختار جدول تصمیم گیری
۵۴۸	شکل ۱۶-۵ جدول تصمیم گیری نتایج



رب ادخلنی مدخل صدق و اخرجنی مخرج صدق وجعل لی من لدنک سلطاناً نصیراً.

حدود یک سال و نیم قبل بود (شاید هم کمی پیشتر) که مطلع شدم آقای دکتر معینی، کتاب حاضر را از انگلستان به همراه آورده و از آقای پرسمن نیز (گویا) مجوز ترجمه را دریافت نموده است. در جلسه ای با حضور ایشان و مهندس کیانی کتاب را سه بخش نمودیم و بر آن شدیم که تا آخر تابستان ۸۰ آماده طبع نماییم. نگارنده بخش خود را (تقریباً) تکمیل نمود ولی قسمت های مربوط به دو بزرگوار به دلیل گرفتاریهای آموزشی و پژوهشی شان آماده نگردید.

با این تصور که ترجمه اندکی از بخشهای مربوط به ایشان کار را به سامان نماید، فعالیت ادامه یافت. تا هنگام که معلوم شد ایشان از انجام این امر انصراف حاصل نموده اند و به ناچار و البته از سر دلتنگی کار را به شخصه پیش بردم و البته در این میان از نظرات بسیاری از دوستان و صاحب نظران بهره بردم. مطابق عادت مالوف پیشین تمام شکلها، جداول و نمودارها باید ترجمه می شد و نه پویش. و نیز از ابتدا به نظمی نگو نرسیم می گردید. حاشیه هایی از ارجاع به سایتها تا فصلهای همین کتاب و سخنان حکیمانه و پرسشهای متداول و نکات کلیدی بسیاری، بر کناره صفحات کتاب نقش بسته بود که ترجمه آنها البته نه به اندازه کتاب و شکل هایش ولی بسیار زمان برد. ارجاعات در تمام فصلها مشخص شده و در متن برگردان شاخص زده شد. کلمات تخصصی و نام افراد و مخفف ها همه در زیرنویس و پانویس ها درج گردیده تا خواننده محترم به اصل آنچه آمده وقوف یابد تا از برگردان، همان استنباطی را داشته باشد که مولف عزیز اراده نموده است. کار پس عظیم بود و وقت کم. نه بیم رقابت که دغدغه شکست در انجام تکلیف تا آخرین روزها صاحب این خامه را می آزد. از طرفی نه شایسته بود و نه ممکن که بخشهایی حذف شوند زیرا ارجاعاتی از بخشهای دیگر و فصلهای دیگر در جای جای متن و حاشیه ها به چشم می آمد و از طرفی زمان نیز به سرعت سپری می گردید. و حتی بیم آن می رفت که نسخه جدید جایگزین گردد.

در این میان من پاسخی بر سوالی قدیم یافتیم که چرا در طول این زمان طولانی کتابی به این اهمیت که مرجع اصلی دروس مهندسی نرم افزار و تحلیل و طراحی سیستم های اطلاعاتی است، مهجور مانده و ترجمه نگردیده است. جواب، سختی کار بود و حجم آن.

باری کار ترجمه از روی کپی کتاب متعلق به دکتر معینی عزیز به پیش می رفت که نسخه آمریکایی کتاب، مطابق معمول و فارغ از دغدغه های "حق نسخه برداری" آفست شده و در دسترس همگان قرار گرفت. استفاده از این کتاب و ترجمه برخی شکلها و مطالب از کتاب فوق که به هر روی مناسب تر از کپی کتاب نخست بود، مشکلی جدید به وجود آورد که درد کهنه را به فراموشی سپرد.

برخی مباحث و حتی شکلها و ارجاع ها تطابق نداشتند و این امر ناسازگاری در متن برگردان را ایجاد می نمود. نسخه بریتانیایی مباحث خادم و مخدوم (Client-Server) را به طور مفصل پوشش داده بود که در نسخه آمریکایی اشاره ای به آن شده بود و نه بیشتر، یا شکلهایی که به کلی تغییر یافته بودند و از این قبیل مشکلات که خود، زمان تحویل نهایی را با تاخیر مواجه می ساخت.

در ترجمه کتاب مشکل مهم دیگری نیز وجود داشت که بسیاری از همکاران کم و بیش با آن مواجهند و آن انتخاب واژگان مناسب تخصصی و نیمه تخصصی بود. کتابچه مهجور فرهنگستان که گویی از سر تکلف و نه تکلیف و شاید برای رفع تکلیف نگاشته شده، دردی دوا نمی نمود که واژگان ساده و ابتدایی نیز قابل دفاع نبودند حتی برگردان رایانه (به جای کامپیوتر) نیز پس از مدتها جایی جای باز ننموده که نظر به رای دارد و اندیشه و این از اتفاق صفت سلیبه کامپیوتر است، هعانی است که کامپیوتر ندارد می دانیم که نقطه ضعفش همان رای و نظر و قضاوت است. و آنچه به کار نیاید در کتابهای تاریخ خوش تر است تا مباحث تخصصی. باری، تا آنجا که وسع مترجم می رسید در یکنواخت نمودن و مستدل بودن واژگان کوشید. Risk را به مخاطره و خطر، Module را به پیمانه و Client/Server را به خادم و مخدوم برگرداندم. Configuration را بیکربندی و Component را نه مولفه که جزء نام نهادم. Planning را طرح ریزی گفتم نه برنامه ریزی تا اگر Program را به برنامه برگردانیم با طرح که برگردان Plan است قابل تمیز باشد. Coding را برنامه نویسی و Programming را برنامه سازی نام نهادم که ایندو را تفاوتی است که خردمندان دانند. Attribute را آنگاه در توصیفی آمده ویژگی گفته ام و هرگاه در مباحث بانکی و پایگاه داده ها مطرح شده، صفت خاصه یا صفت مشخصه نام گرفته و Entity را موجودیت نامیدم به یادگار استاد دانشمند و فداکار و عزیزم روحانی رانکوهی که البته ادبیات او را بسیاری برنتابند. و صدها مورد از این گونه. البته گاه نیز به ضرورتی (که ریشه در انتقال صحیح داشت) اندکی در امانت تصرف نموده ام چون Advice که پند است و نصیحت، ولی آن را توصیه نامیده ام تا از اهمیت و ضرورت رعایت آن نکات، خواننده را مطلع نعایم. (از آن گنه که نفعی رسد به غیر چه باک).

طرح نخست بسیاری از شکلهای کتاب را آقای نیما میرسعیدی تهیه نموده بود که البته سرکار خانم نیره نیکفرجام (که زمانی دانشجویم بوده)، با حوصله و دقت آنان را تکمیل نمود، یا از ابتدا رسم کردند، ایشان تنظیم نهایی کتاب را نیز عهده دار گردید و در طول ماهها تلاش بی وقفه کتاب را به مرحله کنونی رساندند که سزاوار ستایش اند.

برخی تصاویر کتاب توسط آقای حمید حسینی اصلاح شد یا از ابتدا رسم گردید. ایشان پویش و تبدیل به کاراکتر (OCR) ارجاعات و برخی فرمولها را نیز عهده دار بودند. سرکار خانم عصمت داودآبادی تایپ اولیه متون کتاب را انجام دادند و اشخاصی دیگر چون برادریم میثم عزیز و سرکار خانم شادی رستگاران و سرکار خانم هدا حسین زاده (و حتی مادر کوشا و مهربانم) در انجام تنظیمات مختلف کتاب (کم و بیش) یاری ام داده اند و البته جناب دکتر نیکوکار که مدیریت و تولید انتشارات گسترش علوم

پایه را عهده‌دارند، با رویی باز و برخوردی آکادمیک و فارغ از دغدغه‌های بازاری، به چاپ و نشر کتاب همت گماردند، که از همه ایشان و دیگرانی که آشکار و پنهان تأثیرگذار بوده‌اند سپاسگزارم.

به رسم عرف و قانون و اخلاق و ادب از پدید آورنده کتاب، آقای پرسمن کسب اجازه نمودم که برخی از مکتوبات مبادله شده، در ضمیمه این کتاب آمده است. پیشتر یادمان خوبی از این داستان نداشتم که در پاسخم برای مجوز کتاب "هدایت تغییر" مولف آن، قراردادی (چون ترکمانچای) ارسال نمود که بیشتر تجاری بود تا فرهنگی. این بار نیز پس از چند نامه نگاری، استقبال اولیه و احاله به ناشر کتاب، انتشارات مک‌گرو هیل و ادیتور مربوطه و پایان ماجرا!

باری، بر آن بودم که چند سطری نیز برای اهل دل و اهل درد بنگارم، از طراحی سیستم‌های اطلاعاتی مهجور گرفته تا برنامه و بودجه بیمار و شاخص‌های فراموش شده آموزش عالی و تراژدی چاپ و نشر و فرهنگ و شعور فرهنگی. حتی پیش نویس آن را نیز آماده کردم. اما پشیمان شدم چرا که:

هر که را اسرار حق آموختند مهر کردند و دهانش دوختند

در طول سالیان اخیر دانشجویانم، سنگ صبور این درد دل‌ها بوده‌اند و البته بسیار امیدوارم و نو امید که دستی برافشانند و هریک گوشه‌ای بگیرند به اصلاح، که حرم در پیش است و حرامی در پس، اگر رفتی بردی و گر خفتی....

گفتگو آیین درویشی نبود ارنه با تو گفتگوها داشتم

نوید هاشمی طبّا

بهار ۸۲

پیش‌گفتار

نگارش سخنانی چند به عنوان پیش‌گفتار بر کتابی که سالها اگر نه بهترین، که بر فروش ترین کتاب دنیا در حوزه مهندسی نرم افزار بوده، امری است افتخار آفرین و البته حساس. دنیای کنونی دنیای سعی و خطا نیست. زمانی در مقدمه کتابی، بزرگی نگاشت: "این اثر از آن روی قلمی نمودم که هر نوشته ای را یک بار خواندن بیارزد ... " اکنون بر این باوریم که این امر به هیچ روی متصور نیست، چنانکه بهترین ها را نیز در قلمرو خاص، نتوان در عمر مفید تورق نمود، چه رسد به خواندن و به کار بستن.

در انتخاب این کتاب به نظرات بزرگان مراجعه داشتیم. سایتهای مختلف اینترنتی منعکس کننده نظرات صاحب نظران است... دانشمندی آن را "پیشان" خوانده بود و متخصصی "سرشار از حاشیه"، دیگری "دریایی به عمقی اندک" و نظرانی از این گونه البته کم نبود. اما نظر اکثر مطالعه کنندگان و اقبال مهندسين و مرجع دانستن آن در دانشگاه های معتبر، گستره مطالب و در برخی بخشها "عمق قابل توجه آن"، شک و تردید را در خصوص اهتمام به ترجمه آن می زدود و این چنین شد...

مهندسی نرم افزار در کشور ما مهجور مانده است. ساخت سیستمهای علمی که اقتصادی نیز باشد، اینگونه است. چه باید کرد؟ اهل دل و اهل درد می دانند که نرم افزار جرم ندارد، فیزیک ندارد، حجم آن بسته به طریقه بازنمایی و رسانه محملش متغیر می نماید و البته با حجمهای قابل حس دیگر، متفاوت.

استراتژی، تاکتیک، برنامه، تکنیک، اجرا و پشتیبانی همه و همه در دستور کار مهندسی نرم افزارند. مهندسی بدون طراحی بی معنی است. چون مبارزه ای است که تهاجم با امواج انسانی بدور از تکنولوژی و درایت، رویکرد اصلی باشد که به تحقیق، نتیجه ای جز شکست نخواهد داشت. بر آن نیستم که بحثی فلسفی را در این چند سطر ارائه نمایم که ضرورتهایی مهم تر در خصوص کتاب حاضر، وجود دارند.

پرفسور برسمن، نحوه تدریس و مطالعه مطالب و ترتیب و ترتب این امر را خود در مقدمه خویش آورده است. آنچه مترجم ضروری می داند، تاکید بر این نکته است که پس از سالها تدریس مهندسی نرم افزار و تحلیل و طراحی سیستمهای اطلاعاتی، این حقیقت آشکار شده که بدون همکاری و دل دادن دانشجویان و کار گروهی آنان بر پروژه ای گروهی (که شورای عالی برنامه ریزی نیز توصیه نموده) هیچ نتیجه ای از مباحث حاصل نخواهد شد و آنچه به دست خواهد آمد، البته هیچ است.

صاحب این قلم، تقسیم مطالب و مباحث مهندسی نرم افزار را حداقل در دو حوزه مدیریت پروژه های نرم افزاری و توسعه نرم افزار، توصیه می نماید. که در بخش نخست، مطالبی چون طرح ریزی و مدیریت و کنترل پروژه و تخمین و برآورد منابع و مدیریت ریسک و ... قابل طرح است و در بخش دیگر، رهیافت های مهندسی سیستم از مباحث شی گرا گرفته تا مهندسی مجدد و مباحث تضمین کیفیت

می‌تواند مورد بررسی قرار گیرد. مبانی بحث‌های فوق می‌تواند توسط اساتید عزیز ارائه شود و دانشجویان پروژه و سمیناری برای تکمیل و تعمیق مطالب، بر عهده گیرند.

کتابی که در اختیار دارید، به ضرورت حجم در دو جلد منتشر گردید. به تحقیق نمی‌توان گفت که مباحث مهندسی نرم افزار ۱ در جلد نخست واقع شده هر چند مطالب مربوط به درس مهندسی نرم افزار پیشرفته را در بخش پنجم کتاب و در جلد دوم می‌توان یافت. بهر روی ارجاعات بخشها و سلسه مراتب بحثها، خصوصاً مطالعه موردی که در طول کتاب به آن اشاره گردیده باعث گردید که امانت در ترتیب فصلها کاملاً رعایت شود. و به همین علت هم دوره دوجلدی به طبع رسید.

در آخر ذکر این نکته را ضروری می‌دانم که علی‌رغم کوشش به عمل آمده و حساسیت‌هایی که مترجم در برگردان متن به خرج داده که گاه ادبیات روزانه و آشنای او را نیز به چالش کشیده است، خود را و محصول عمل خود را از خطا، بری ندانسته و هر نقدی را از هر بزرگی و هر عزیزی خواه محتوایی و خواه شکلی، بر دیده منت می‌نهد. نظرات مولوی و ارشادی خود را به سایت مترجم : WWW.hashemitaba.net ارسال فرمایید.

غرض نقشی است کزما بآزماید که دنیا را نمی‌بینم بقایی

نوید هاشمی‌طبا

مقدمه

هنگامی یک نرم افزار کامپیوتری موفق قلمداد می گردد که تمام نیازمندیهای کاربرانش را برطرف سازد و این بدان معناست که مدت زمانی طولانی بدون عیب و نقص کار کرده یا اصلاح و حتی اعمال تغییرات در آن ساده باشد و بتواند و باید بتواند که امور را به گونه ای مناسب تر تغییر دهد. اما وقتی نرم افزاری با موفقیت همراه نیست، یعنی کاربر از آن ناراضی است، مستعد خطاست، تغییر آن مشکل است، حتی کار کردن با آن سخت و طاقت فرسات.

رخداد هر اتفاق نامطلوب، ممکن است از اینرو ما همگی می خواهیم نرم افزاری بسازیم که بهتر عمل کند و از اتفاقات بدی که در سایه فعالیت های شکست خورده، کمین کرده اند، بپرهیزد. موفقیت در طراحی و ساخت نرم افزار نیاز به انضباط^۱ و نظم دارد و البته رهیافتی^۲ مهندسی.

در طول ۲۰ سال از اولین چاپ این کتاب، طراحی نرم افزار از یک ایده ناشناخته که تعداد نسبتاً اندکی از افراد متعصب از آن بهره می بردند، به یک دیسیپلین^۱ قانونمند در طراحی و مهندسی تبدیل شده است. امروزه این امر بعنوان یک موضوع ارزشمند در تحقیقات جدی، بررسی های آگاهانه و مشاجرات و بحث های جنجالی مشخص گردیده است. در سراسر این صنعت، مهندسین نرم افزار جایگزین برنامه ساز شده است و به صورت یک عنوان شغلی دارای اولویت در آمده است. مدل های پردازش نرم افزار، روشهای مهندسی نرم افزار و ابزارهای نرم افزاری در طول طیف وسیعی از برنامه ها، کاربردی صنعتی به گونه ای موفقیت آمیز پذیرفته شده اند.

گرچه مدیران و متخصصان نیاز به رهیافتی منظم تر را تشخیص داده اند، اما درخصوص کاربرد رهیافت ها نیز به بحث و جدل مشغولند. هنوز نیز بسیاری از افراد و شرکتها نرم افزار را به طور آشفته و نامنظم تولید می کنند هرچند سیستمهایی می سازند که خدماتی به اکثر فناوریهای پیشرفته امروزی ارائه می کنند. بسیاری از افراد حرفه ای و دانشجویان از روشهای مدرن اطلاعی ندارند و در نتیجه کیفیت نرم افزار تولید ما دچار نقیصه شده و اتفاقات بدی روی می دهد. علاوه بر آن، بحث و تناقض در مورد ماهیت حقیقی رهیافت مهندسی نرم افزار ادامه دارد. وضعیت مهندسی نرم افزار مطالعه تعارضات است. هرچند نگرشها تغییر کرده، پیشرفت حاصل شده، اما قبل از بلوغ تمام و کمال نظم و انضباط هنوز کار زیادی باید انجام گیرد.

چاپ پنجم کتاب مهندسی نرم افزار یک رهیافتی تخصصی، به منظور ارائه خدمات در جهت راهنمایی ارتقاء نظم و انضباط در مهندسی است. چاپ پنجم نیز مانند چاپ چهارم که بیش از آن بوده، هم برای دانشجویان است و هم متخصصین و دارای یک راهنمای تخصصی برای مصارف صنعتی و یک مقدمه برای

^۱. discipline

^۲. Approach

دانشجویان سطوح بالای کارشناسی یا سال اول کارشناسی ارشد است. قالب و سبک چاپ کتاب دچار تغییرات زیادی شده و کتاب حاضر را برای خواننده دلیزیرتر و مطالبش را ساده‌تر ساخته است.

چاپ پنجم تا حد قابل توجهی، چیزی بیشتر از یک نمونه به روز شده ساده خواهد بود. این کتاب آنگونه اصلاح شده که با توسعه گسترده و چشمگیر این حوزه تطابق یافته و بر روشهای مهندسی نرم‌افزار جدید و مهم تأکید نماید. علاوه بر آن یک سایت جامع شبکه‌ای توسعه یافته تا محتوای کتاب را تکمیل کند.

وب سایت ای که من آنرا Sepaweb نامیدم، در این آدرس خواهد بود.

<http://www.mhhe.com/pressman>

سایت مذکور یک سری منابع مهندسی نرم‌افزار را که برای اساتید و دانشجویان و متخصصان صنعتی سودمند می‌باشد، ارائه می‌دهد.

۳۲ فصل چاپ پنجم در پنج بخش سازماندهی شده است. این کار به منظور قسمت قسمت کردن عناوین و کمک به اساتیدی است که وقت ندارند کل کتاب را در یک ترم تدریس نمایند. بخش اول، محصول و فرآیند، مقدمه‌ای است بر محیط مهندسی نرم‌افزار. در این بخش سعی شده موضوع مورد بحث معرفی شود و از همه مهمتر آنکه، مفاهیمی ارائه شود که برای فصول بعدی دانستن آنها ضروری می‌نماید. بخش دوم، مدیریت پروژه‌های نرم‌افزاری، دارای عناوینی است که به افرادی مربوط می‌شود که طرح‌ریزی، مدیریت و کنترل پروژه توسعه نرم‌افزاری را انجام می‌دهند. بخش سوم، روشهای متعارف برای مهندسی نرم‌افزار، نشان دهنده روشهای تحلیل کلاسیک و سنتی برای آزمون و تست است که برخی آن را بعنوان مکتب قراردادی مهندسی نرم‌افزار در نظر می‌گیرند. بخش چهارم، طراحی مهندسی نرم‌افزار شیء‌گرا (Object-Oriented) نمایانگر روشهای شیء‌گرا در طول کل فرآیند مهندسی نرم‌افزار است. از جمله تحلیل، طراحی و آزمون. بخش پنجم یا عناوین پیشرفته در مهندسی نرم‌افزار، نمایانگر فصول تخصصی است که روشهای رسمی، طراحی مهندسی نرم‌افزار اتاق پاک، مهندسی نرم‌افزار مبتنی بر جزء، طراحی مهندسی خادم/مخدوم، مهندسی وب، طراحی مجدد و مهندسی نرم‌افزار به کمک ابزار کامپیوتری را دربر می‌گیرد.

این ساختار پنج بخشی در چاپ پنجم، استاد را قادر می‌سازد عناوین مطلوب را براساس زمان در اختیار و نیاز دانشجو گروهبندی کند. یک دوره یک ترمی کامل باید در مورد یک یا دو بخش از کل پنج بخش بوجود آید. قبلاً طرح‌ریزی یک دوره ممکن است به گونه‌ای باشد که روی بخش سه یا چهار تأکید کند، یا دوره‌ای دیگر فصولی را از بخشهای سه و چهار و پنج دربر گیرد. در یک دوره دیگر مدیریت بخشهای یک و دو را مورد تأکید قرار می‌دهد. با سازمان دادن چاپ پنجم به صورت یاد شده، تلاش کرده‌ام تعدادی گزینه تدریسی در اختیار استاد قرار دهم.

کار من روی چاپ پنجم این کتاب طولانی ترین کار بر روی یک پروژه فنی مستمر در تمام طول حیاتم بوده است. حتی وقتی از نوشتن دست می کشیدم، اطلاعات استخراجی از مقالات و گزارشات را تحت تلفیق و سازماندهی قرار می دادم. از همین روی، باید از بسیاری نویسندگان کتابها، مقالات و گزارشات بعلاوه نسل جدید وسایل کمکی الکترونیکی چون گروه خبری، نویسندگان خبر الکترونیکی و شبکه گسترده جهانی (WWW) تشکر کنم زیرا دیدگاهها، ایده ها و توصیه هایی را در طول ۲۰ سال گذشته در اختیارم نهاده اند که در بسیاری از صفحات این کتاب به آنها اشاره کرده ام. همگی آنها برای تکمیل سریعتر این کتاب کمک های زیادی نموده اند که مستحق سپاسگزاری اند. همچنین مايلم از خوانندگان مجدد این کتاب تشکر نمایم. پیشنهادات و انتقادات آنها بسیار ارزشمند بود. سپاسگزاری خاص خود را نثار بروس ماگزیم از دانشگاه دیربون، میشیگان می نمایم، هم او که در ایجاد سایت شبکه مربوط به این کتاب مرا کمک نمود بروس مسئول بسیاری از طراحیها و مفاهیم آموزشی کتاب است. مطالب چاپ پنجم بوسیله اساتید حرفه ای حوزه صنعت، اساتید دانشگاه و دانشجویانی تدوین شده که از چاپ های قبلی این کتاب استفاده کرده و وقت زیادی صرف ارائه پیشنهادات، انتقادات و ایده های خود نموده اند. از تک تک شما نیز متشکرم. علاوه بر آن یک تشکر شخصی دارم از بسیاری از کاربران صنعتی خود در سراسر جهان، آنان که مطمئناً بیش از آنکه من به آنها بیاموزم، آنها به ما می آموزند.

با تکمیل چاپ های این کتاب، پسران من، مایو و مایکل از پسرانی جوان به مردانی بزرگ تبدیل شدند. بلوغ آنها، شخصیت و موفقیتشان در جهان واقعی برای من یک الهام و قوت قلبی بوده است. هیچ چیز چون این مسئله برای من افتخار آفرین نبوده است. و نهایتاً برای عشقم باربارا و با تشکر فراوان از او که هنوز هم مشوق من، برای چاپ دیگری از این کتاب است.

راجر اس. پرسمن

کتاب راجر پرسمن در حوزه مهندسی نرم افزار کتابی ممتاز و عالی است: من همواره گستردگی مطالب و توانایی نویسنده را در توصیف جزئیات مشکل را درک کرده و آن را ستوده ام. از اینرو هنگامی که انتشارات مک گرو هیل از من درخواست یک نسخه اروپایی از چاپ پنجم را نمود، بسیار مفتخر شدم. این سومین اقتباس است و حاوی مطالب چاپ پنجم آمریکایی و مطالبی است که من برای نسخه اروپایی نوشته ام.

چندین حوزه وجود دارد که در آن کتاب دارای مطالب اضافه تری است:

- مطالب بسیاری در مورد روشهای رسمی توسعه نرم افزار وجود دارد. این فنون در آلمان و انگلستان پایه گذاری شدند و به بخشی از بسته های نرم افزاری مهم مورد استفاده دائمی مهندسين تبدیل گردیدند که از آنها در توسعه سیستم های مهم و یکپارچه و هماهنگ امنیتی، استفاده می شود.

• من بخشی را درخصوص الگوهای طرح‌ریزی گنجانده‌ام. اینها معمولاً بصورت ساختارهای بسیار کوچک روی می‌دهند که بارها و بارها در سیستمهای شیء‌گرا استفاده شده و به مثابه شابلونهای طراحی هستند که می‌توان آنها را دوباره و چند باره استفاده نمود. من به اروپا سفر می‌کنم و مرتباً با شرکتهایی که واقعاً با چنین فناوری کتابخانه‌ای را به شما می‌دهند، برخورد دارم.

• من متریک را مورد توجه قرار دادم و بویژه استفاده از GQM بعنوان یک روش توسعه متریک. هرچه مهندسی نرم افزار منظم تر می‌شود، این فناوری یکی از پایه‌ها و بنیان‌های آن می‌شود. متریک یک مقیاس اروپایی است و امیدوارم فضایی که به آن اختصاص دادم این مسئله را منعکس کند.

• من مطالبی را براساس UML با زبان مدلسازی یکپارچه وارد کردم که منعکس کننده افزایش شدید کاربرد این مجموعه در اروپای غربی است. گمان دارم در طول سه، چهار سال آینده این داستان، استاندارد توسعه نرم‌افزار شود.

• من بر توسعه سیستم توزیع شده با استفاده از زبان برنامه سازی 'جاوا' تأکید ورزیده‌ام تا آنجاکه بعضی از برنامه‌های مربوطه را تشریح نموده‌ام. اکنون که اینترنت و تجارت الکترونیک در اروپا در حال تحول کامل می‌باشند، شرکتهای بیشتر و بیشتری جهت توسعه برنامه‌های کاربردی توزیع شده خود به سوی فنون مهندسی نرم‌افزار روی خواهند آورد. این کتاب بازتاب این مسئله است.

• مطالبی درمورد روشهای امنیتی و مهندسی گنجانده‌ام. با استفاده از اینترنت (شبکه جهانی) هر مهندس نرم‌افزار باید چیزهای بیشتری درمورد فنونی مثل امضاء رقعی و رمزنگاری بداند.

• در بخشهای انتهایی کتاب من بر استفاده از برنامه‌های کاربردی تجارت الکترونیک تأکید کرده‌ام تا فناوری مربوطه را تشریح نمایم.

در این مورد دو بخش وجود دارد: یک وب سایت بزرگ آمریکایی که بوسیله دکتر پرسمن ارائه شده بود و یک سایت ورودی که آدرس آن در کتاب در پایان فصول ارائه شده است. این سایت ورودی حاوی مطالب مربوط به چاپ اروپایی و مرتبط با سایت آمریکایی است. هر دو سایت به هم مرتبط شده در برگیرنده شبکه‌های کوچکتری هستند که دارای منابع دانشجویی، منابعی برای اساتید و نیز منابعی برای افراد حرفه‌ای است.

منابع دانشجویی شامل یک راهنمای تحقیق است که نکات اصلی کتاب را خلاصه نموده، به همراه یک سری خودآزمون که شما را قادر می‌سازد میزان درک خود را از مطالب محک زنید، صدها رابط با منابع مهندسی نرم افزار، یک تحقیق موردی، مطالب مکمل و یک صفحه پیام که شما را قادر می‌سازد با دیگر خوانندگان صحبت کنید.

منابع اساتید شامل یک راهنمایی استاد، مجموعه‌ای از نکات مفید ارائه شده براساس کتاب، مجموعه‌ای از سؤالات که می‌توانند برای تکلیف و امتحان مورد استفاده واقع شوند. یک سری ابزارهای

کوچک (ابزارهای ساده مهندسی نرم افزار که برای کاربرد کلاسی مناسبند)، یک ابزار شبکه‌ای که به شما امکان می‌دهد سایت ویژه رشته خود را در شبکه ایجاد کنید و یک صفحه پیام که شما را قادر می‌سازد با دیگر اساتید ارتباط برقرار کنید.

منابع حرفه‌ای شامل مدارکی برای فرایندهای مهندسی نرم افزار، لیست‌های کنترلی (چک‌لیستها) برای فعالیتهایی چون بازمینی، رابطهایی با فروشندگان حرفه‌ای ابزار، صدها رابط با منابع مهندسی نرم افزار، اطلاعاتی درمورد بسته‌های ویدئویی "پرسمن" و پیشنهادات و مقالات صنعتی در مورد تعدادی از عناوین و موضوعات مهندسی نرم افزار است. کتاب مهندسی نرم افزار یک کتاب ممتاز است. امیدوارم که مطالب اضافه ای که برای خواننده اروپایی ارائه کرده ام، ضروری و به درد بخور باشد.

دارل اینس

پنجمین چاپ کتاب مهندسی نرم افزار: یک رهیافت متخصصین، مجدداً برنامه‌ریزی و طراحی شده تا تجربه مطالعاتی شما را افزایش داده و رابط‌های یکپارچه‌ای با سایت وب <http://www.mhhe.com> /pressman مهیا کند. این سایت مشتمل بر اطلاعات تکمیلی بسیار ارزشمندی برای خوانندگان کتاب و یک مجموعه منابع برای اساتیدی است که این کتاب را به عنوان یک کتاب درسی برای کلاس خود برگزیده‌اند.

در سراسر این کتاب شما با شمایی (ایکون‌هایی) در حاشیه مواجه می‌شوید که باید مفهوم آنها را بدانید. این مفاهیم در زیر تشریح شده است:

ایکون ارجاع به وب اشاراتی مستقیم به سایت‌هایی دارد که برنامه‌های نرم افزاری مرتبط را دربر دارند.



ایکون Website نشان دهنده اطلاعات بیشتری در مورد عنوان مورد بررسی است که در سایت وب SEPA می‌باشد.



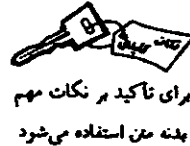
ایکون چک لیست چک لیست‌های دقیقی را نشان می‌دهد که یاری‌گر شما در دسترسی به کار مهندسی نرم‌افزار مورد استفاده شما است و اینکه محصولات کاری خود را بهتر ببینید.



ایکون مستندات طرح‌های اسنادی دقیق، توضیحات و توصیف‌هایی و نمونه‌هایی موجود در سایت وب SEPA را به شما نشان می‌دهد.



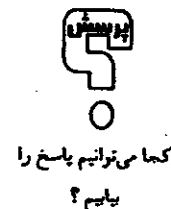
ایکون نکات کلیدی به شما در فراگیری سریع نکات مهم و کلیدی کمک می‌کند.



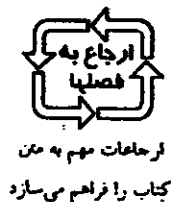
ایکون توصیه هدایتگر عملی شما در کمک به تصمیم‌گیری درست می‌باشد، به گونه‌ای که از مشکلات معمول در حین استفاده از نرم‌افزار بپرهیزید.



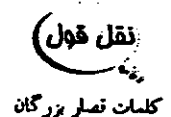
ایکون پرسش یک سری سؤالات متداول را ارائه می‌کند که پاسخ آنها در متن وجود دارد.



ایکون ارجاع به فصلها به قسمت دیگری از کتاب اشاره می‌کند که اطلاعاتی درخصوص مبحث کنونی را می‌توان آنجا یافت.



ایکون نقل قول نقل قول‌هایی جالب را که به نوعی مرتبط به موضوع مورد بررسی می‌باشند، بیان می‌دارد.



بخش اول محصول و پردازش

در این بخش از کتاب "مهندسی نرم افزار - رهیافتی برای یک اهل فن" شما، درخصوص محصولی که قرار است طراحی شود و فرآیندی که چارچوب ساخت محصول را تهیه می کند، مطالب و فناوری‌هایی چند را، خواهید آموخت. سؤالات زیر در فصلهای پیش رو مورد بررسی قرار می گیرد:

- یک نرم افزار کامپیوتری، حقیقتاً چیست؟
 - چرا تلاش داریم که سیستم‌های با کیفیت را برپایه کامپیوتر بسازیم؟
 - چه رمز و رازهایی هنوز درمورد کامپیوتر وجود دارد؟
 - فرآیند نرم‌افزاری چیست؟
 - آیا راه حل کلی برای دسترسی به کیفیت یک فرآیند وجود دارد؟
 - مدلهای فرآیند خطی و تکرار شونده از چه لحاظ متفاوتند؟
 - نقاط ضعف و قوت آنها کدامند؟
 - چه مدل‌های نرم‌افزاری پیشرفته ای برای کار مهندسی نرم‌افزار پیشنهاد شده است؟
- وقتی به سؤالات فوق پاسخ داده شود، شما آمادگی درک جنبه‌های مدیریتی و فنی رهیافت مهندسی توصیف شده در ادامه این کتاب را کسب خواهید نمود.

محصول

فصل ۱

مفاهیم کلیدی (مرتب بر حروف الفبا)

استفاده مجدد ، افسانه ، تاریخچه ، زوال و کهنگی ، طبقه بندی برنامه های کاربردی ،
مشخصات نرم افزار ، منحنی های شکست ، مونتاژ مبتنی بر اجزاء ، مهندسی نرم افزار

KEY CONCEPTS

Application categories , component-based assembly , failure curves , history ,
myths , reuse , Software characteristics , Software engineering , wear

نگاه اجمالی

نرم افزار کامپیوتری چیست؟ محصولی است که مهندسين نرم افزار آن را طراحی و تولید می کنند.
شامل برنامه هایی است که در یک کامپیوتر یا هر اندازه و ساختاری اجرا می شود، مستنداتی هستند که کد
برنامه ها را و شکلهای و چارتهای را دربر گرفته و نیز داده هایی هستند که متشکل از لرقام و متون می باشند، اما
مشمول بر نمایش تصویری، ویدئویی و اطلاعات نیز هستند.

چه کسی آنها می سازد؟ کار، کار مهندسين نرم افزار است و در واقع هر آنکس که در جهان
صنعت از آن به طور مستقیم یا غیرمستقیم بهره می برد.

چرا این ابزار مهم است؟ از آنرو که تقریباً بر هر جنبه از زندگی ما اثرگذار است و در تجارت بسیار
گسترده شده است. همچنین در فرهنگ و فعالیتهای روزمره ما نیز مورد استفاده دارد.

مراحل کار چیست؟ شما نرم افزار کامپیوتری را مثل هر محصول موفق دیگر تولید می کنید یعنی با
به کارگیری فرآیندی که منجر به نتایج بسیار چشمگیری می شود که نیازهای افرادی که استفاده کنندگان
آن محصول هستند را، رفع می سازد. شما از یک رهیافت مهندسی نرم افزار استفاده می کنید.

حاصل کار چه خواهد بود؟ از دیدگاه مهندس کامپیوتر محصول به دست آمده برنامه‌ها، مستندات و داده‌هایی است که نرم‌افزار کامپیوتر را تشکیل می‌دهد. اما از دیدگاه کاربر، محصول عمل، اطلاعات بدست آمده است که به نوعی جهان کاربر را بهتر می‌سازد.

چگونه از درستی کار خود اطمینان حاصل نمایم؟ باقی کتاب را بخوانید، ایده‌هایی را انتخاب کنید که در نرم‌افزاری که می‌سازید، قابل اجرا هستند و از آنها در کار خود استفاده کنید. هشدارها یک دهه قبل از حادثه شروع شد. کمتر از دو سال به مهلت مقرر مانده بود که رسانه‌ها داستان را برملا کردند. سپس مأمورین دولتی نگرانی خود را ابراز نمودند، رؤسای تجاری و صنعتی مقادیر متنابهی سرمایه‌گذاری را متعهد شدند و نهایتاً هشدار مخوفی درمورد فاجعه قریب‌الوقوع به اذهان عمومی رخنه کرد. نرم‌افزار به دلیل مشکل و نقیصه معروف Y2K، رو به شکست بود، شکستی که جهان را متوقف می‌نمود.^۱

در طول ماههای آخر سال ۱۹۹۹ ما همگی چشم نگران و سرگردان بودیم. من نمی‌توانستم به یک بند از صفحه اول چاپ چهارم کتاب نیاندیشم، در این بند به گونه‌ای پیشگویانه آمده بود: نرم‌افزار کامپیوتری به صورت یک نیروی پیش‌برنده در آمده است. نرم‌افزار موتوری است که تصمیم‌گیری تجاری را به تحرک وامی‌دارد و به عنوان پایه و اساس تحقیقات علمی مدرن و حل مشکلات مهندسی عمل می‌نماید. نرم‌افزار کامپیوتری عامل اصلی در ایجاد تمایز میان محصولات و خدمات مختلف است. نرم‌افزار در سیستمهایی از همه نوع همچون حمل و نقل، پزشکی، ارتباط از راه دور، نظامی، فرآیندهای صنعتی، محصولات دفتری و تفریحی نهفته است. این فهرست تقریباً بی‌پایان است. در جهان مدرن تقریباً گریزی از آن نیست و با حرکت ما به سوی قرن ۲۱، پیش‌برنده ما به سوی پیشرفتهای جدید در هر امری خواهد بود، از آموزش ابتدایی گرفته تا مهندسی ژنتیک.

در طی پنج سالی که از چاپ چهارم این کتاب می‌گذرد، نقش نرم‌افزار به عنوان نیروی پیش‌برنده کاملاً مشخص شده است. اینترنتی که از نرم‌افزار مشتق شده، اقتصاد پانصد میلیارد یورویی خود را بوجود آورده است. باتوجه به جدید بودن نمونه اقتصادی، سرمایه‌گذاران وال استریت به تدرت از اعتبارات میلیارد

۱. مشکل سال ۲۰۰۰ این بود که فیلد تاریخ در اکثر نرم‌افزارها دو رقمی در نظر گرفته شده بود و با شروع سال دوهزار میلادی صفر شده، قابلیت اطمینان نرم‌افزارها را با اشکال جدی مواجه می‌ساخت. مخصوصاً آن نرم‌افزارهایی که از تاریخ در فرمولها بهره برده بودند ممکن بود با صفر شدن مخرج کسرها، با اشکال تقسیم بر صفر مواجه شوند که اصولاً خطایی غیرقابل پوشش با پیامدی وخیم بود. مترجم.

دلاری به شرکتهای کامپیوتری پرداختی نمودند، پیش از آنکه کارشان به فروش برود. صنایع جدیدی از نرم افزار بوجود آمده اند و صنایع قدیمی که با آن تطابق نیافته اند، اکنون تهدید به نابودی شده اند. دولت ایالات متحده علیه بزرگترین شرکت در صنعت نرم افزار اقامه دعوا نموده است. همانگونه که در دوره های پیشین نیز وقتی می خواست شیوه های انحصار طلبانه را در صنایع نفت و فولاد از بین ببرد، اینکار را انجام داد.

تأثیر نرم افزار بر جامعه و فرهنگ ما بسیار ژرف و عمیق است. با افزایش میزان اهمیت آن، جامعه نرم افزاری مرتباً تلاش می کند تا فناوریهای را توسعه بخشند که ساختن برنامه های کامپیوتری را بصورت ساده تر، سریعتر و ارزان تر در آورد.

این کتاب بیانگر چارچوبی است که میتوان از آن برای تولید برنامه های نرم افزاری کامپیوتری استفاده نمود و البته توسط افرادی که آن را به درستی انجام دهند. فناوری در برگزیده یک فرایند، مجموعه ای از روشها و یک مجموعه ابزارهایی است که، آنرا مهندسی نرم افزار می نامیم.

نقل قول

ایده ها و یافته های فنی و تکنولوژیکی موتور محرکه رشد اقتصادی می باشند.
وال استریت ژورنال

۱-۱ نقش تکاملی نرم افزار

امروزه، نرم افزار نقش دوگانه ای دارد. نرم افزار خود یک محصول است و در عین حال وسیله ای برای ساخت و تحویل محصول به حساب می آید. نرم افزار به عنوان یک محصول، پتانسیل محاسباتی نهفته در سخت افزار کامپیوتر یا به صورت کلی تر شبکه ای از کامپیوترها را ارائه می دهد که از طریق سخت افزار محلی، قابل دسترسی هستند. این نرم افزار چه در یک تلفن سلولی قرار داشته باشد یا درون یک کامپیوتر بزرگ، یک انتقال دهنده اطلاعات محسوب می گردد. یعنی اطلاعاتی را تولید می کند، سازمان می دهد، تحویل می گیرد، نمایش یا انتقال می دهد که ممکن است به سادگی یک بیت باشد یا به گونه ای چند رسانه ای، پیچیده. اگر از آن به عنوان وسیله انتقال دهنده محصول استفاده شود، به صورت پایه و اساس کنترل کامپیوتر (سیستم عامل)، تبادل اطلاعات (شبکه ها) و ایجاد و کنترل برنامه های دیگر (ابزارهای نرم افزاری و محیط آن) استفاده خواهد شد.



نرم افزار هم محصول است و هم وسیله ای برای انتقال و حمل محصول

در طول گذشت ۵۰ سال، نقش نرم افزار کامپیوتر، تغییرات چشمگیری داشته است. پیشرفتهای گسترده در عملکرد سخت افزار، تغییرات شگرف در ساختار کامپیوترها و محاسبات، افزایش شدید حافظه و ظرفیت نگهداری اطلاعات و یک سری گزینه های ورودی - خروجی متنوع و هیجان انگیز، همگی باعث پیچیده تر کردن سیستم های مبتنی بر کامپیوتر شده اند. پیچیدگی و تودرتویی می تواند باعث نتایج خیره کننده ای در هنگام موفقیت سیستم شود اما برای کسانی که این سیستم های پیچیده را تولید می کنند، مشکلات عدیده ای بوجود می آید.

کتابهای معروفی که در طول دهه ۷۰ و ۸۰ چاپ شدند دیدگاه مؤثر و قدیمی در مورد برداشت در حال تغییر، از کامپیوتر و نرم‌افزار و تأثیر آنها بر فرهنگ ما، ارائه می‌دادند. آزیورن [OSB79]^۱ یک انقلاب صنعتی جدید را توصیف می‌کند. تافلر [TOF80]^۲ ایجاد علم میکرو الکترونیکها را بخشی از موج سوم تغییر در تاریخ بشر خواند و نیزبیت [NAI89]^۳ انتقال از یک جامعه صنعتی به یک جامعه اطلاعاتی را پیش‌بینی نمود. فیگن بام و مک کورداک [FEI83]^۴ بیان کردند که دانش و اطلاعات (که تحت کنترل کامپیوترهاست) نقطه کلونی قدرت در قرن بیست و یکم خواهد بود واستول [STO89]^۵ بیان داشت که جامعه الکترونیکی که بوسیله شبکه‌ها و نرم‌افزار تشکیل شده کلید تبادلات علمی در سراسر جهان خواهد بود.

با شروع دهه ۹۰ تافلر [TOF90]^۶ یک تغییر قدرت را توصیف کرد که در آن ساختار قدیمی قدرت (یعنی قدرت دولت، آموزش، صنعت، اقتصاد و ارتش) بصورت کامپیوتر و نرم‌افزار درمی‌آید که منجر به «آزادسازی یا دموکراتیزاسیون دانش» می‌شود. یوردان [YOU92]^۷ اظهار نگرانی می‌کند که شرکتهای آمریکایی ممکن است در نرم‌افزارهای مربوط به تجارت، لبه رقابت را از دست بدهند و پیش‌بینی کرد که برنامه‌ریزی آمریکایی نزول و سقوط خواهد کرد. همر و چمپی [HAM93]^۸ بیان کردند که فناوریهای اطلاعاتی نقش محوری را در برنامه‌ریزی مجدد شرکت ایفا خواهند نمود. در اواسط دهه ۹۰ گستردگی کامپیوترها و نرم‌افزارها باعث انتشار یک سری کتاب توسط افراد ضدفناوری جدید شد.^۹ این نویسندگان با تأکید بر بحق بودن نگرانیهایشان و بدون توجه به اثبوه مزایایی که هم‌اکنون مشخص گردیده است، سعی کردند کامپیوترها را از دور خارج کنند.^{۱۰} [LEV95]

اواخر دهه ۹۰، یوردون [YOU96] چشم اندازهای حرفه نرم‌افزار را مجدداً ارزیابی نمود و "حیات مجدد و رشد" برنامه‌نویسان آمریکایی را اعلام نمود. با افزایش اهمیت اینترنت، معلوم شد که حدس او درست بوده است. با اتمام قرن بیستم، این نقطه تمرکز یک بار دیگر تغییر کرد و این بار متوجه بمب ساعتی Y2K [YOU98b]^{۱۱} شد. گرچه بسیاری افراد، پیش‌بینی محکوم کنندگان Y2K [DEJ98]^{۱۲} و

نقل قول

هنگامیکه به گونه ای عمیق به آینده می اندیشیم، بسیار فراتر از آنچه چشم آدمی می بیند، پیش از جهان بدست می آید و شگفتیهایی که می تواند به وقوع بپیوندد. تئیسون

نقل قول

کامپیوترها بسیاری از امور را تسهیل می سازند ولی بسیاری از آنچه توسط آنها تسهیل می شود زائد و بدون استفاده است. لندی رونی

1. Osborne

2. Toffler

3. Naisbitt

4. Feigenbaum & McCorduck

5. Stoll

6. Toffler

7. Yourdon

8. Hammer & Champy

9. (e.g., Resisting the virtual life)

10. Levy, S.

11. Yourdon, E

و [KAR99]^۱ را بیش از واکنش لازم، برداشت نمودند [NOR98]^۲، اما نوشته‌های معروف آنها باعث رایج شدن نرم‌افزار در زندگی ما شد. امروزه محاسبات متداول، نسلی از وسایل اطلاعاتی را رواج داده که ارتباط گسترده‌ای با شبکه دارند [LEV99]^۴ تا پوشش ارتباطی را در خانه‌ها، دفاتر و بزرگراه‌ها بوجود آورند. نقش نرم‌افزارها بازهم در حال افزایش است.

یک برنامه‌نویس منفرد در عصر پیش، اکنون جای خود را به یک تیم از متخصصان برنامه‌نویسی داده که هریک روی قسمتی از فناوری لازم متمرکز می‌شوند تا یک برنامه کاربردی پیچیده را ارائه نمایند. با این حال برنامه نویسان منفرد پاسخگوی سؤالاتی بودند که در سیستمهای مدرن نیز وجود دارد:

- چرا تکمیل این نرم‌افزار، اینقدر به طول انجامید؟
- چرا هزینه‌های تکمیل کار بسیار بالاست؟
- چرا پیش از تحویل نرم‌افزار به مشتریان، نمی‌توانیم خطاهای موجود در کار را بیابیم؟
- چرا در ارزیابی میزان پیشرفت نرم‌افزار تولید شده، اینقدر با مشکل مواجه هستیم؟

این سؤالات و بسیاری از سؤالات دیگر، در خصوص اهمیت نرم‌افزار و توسعه آن مطرح می‌باشند- اهمیتی که ما را به سوی ممارست مهندسی نرم‌افزار سوق می‌دهد.

۲-۱ نرم‌افزار



چگونه می‌توان تعریف
مناسبی از نرم‌افزار
ارائه نمود؟

در سال ۱۹۷۰، کمتر از یک درصد مردم می‌توانستند بطور معقولی، آنچه را که به معنای نرم‌افزار کامپیوتری بود توصیف کنند. امروزه، اکثر افراد حرفه‌ای و بسیاری از اعضای جامعه فکر می‌کنند که تا حد زیادی مفهوم نرم‌افزار را درک می‌کنند، اما آیا واقعاً اینطور است؟

۱-۲-۱ خصوصیات نرم‌افزار

به منظور شناخت یک نرم‌افزار (و نهائناً شناخت مهندسی نرم‌افزار) بررسی مشخصه‌های نرم‌افزاری که آن را از دیگر چیزهایی که انسان ساخته متمایز می‌سازد، امری مهم است. وقتی سخت‌افزار ایجاد می‌شود، رهیافت خلاقیت انسان، (تحلیل، طراحی، ساخت و آزمون) نهائناً شکل فیزیکی بخود خواهد گرفت. اگر یک کامپیوتر جدید بسازیم طرحهای اولیه، نقشه‌های رسمی مربوط به طراحی و نمونه‌های ساخته شده بصورت یک محصول فیزیکی تکمیل می‌شوند (تراشه‌ها، صفحات مدار، منبع برق و غیره).

1. De Jager, P

2. Karlson, E

3. Norman, D

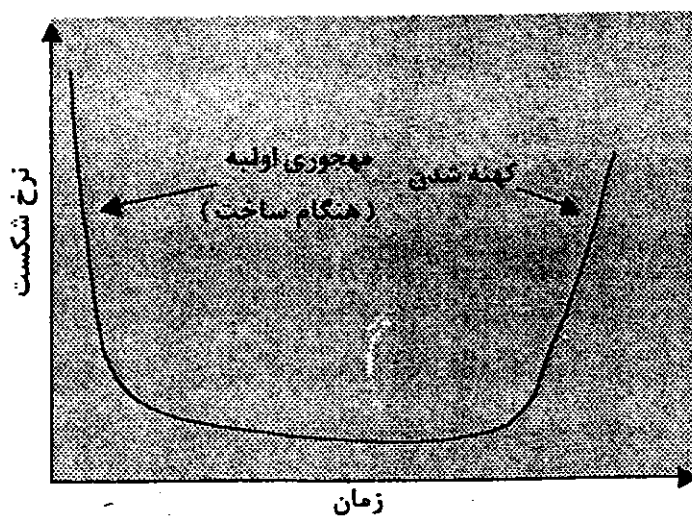
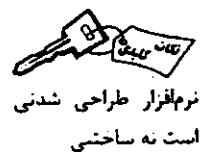
4. Levy, S

نرم افزار بیشتر یک عنصر منطقی است تا یک سیستم فیزیکی. بنابراین دارای مشخصه‌هایی است که تا حد زیادی از مشخصه‌های یک سخت افزار متفاوتند.

۱- نرم افزار توسعه می یابد یا طراحی می شود، اما به مفهوم کلاسیک ساخته نمی شود.

گرچه شباهتهایی بین توسعه نرم افزار و ساخت سخت افزار وجود دارد، اما این دو فعالیت اساساً با یکدیگر متفاوتند. هرچند در هر دوی آنها، کیفیت بالا حاصل طراحی خوب خواهد بود، اما مرحله ساخت در مورد سخت افزار می تواند یک سری مشکلات کیفی داشته باشد که در مورد نرم افزار وجود ندارد یا به راحتی حل شدنی و رفع شدنی خواهد بود. هر دو فعالیت وابسته به مردم هستند اما ارتباط بین افراد متخصص و کار صورت گرفته کاملاً متفاوت است (فصل ۷). هر دو فعالیت مستلزم ساختن یک محصول هستند اما روشها کاملاً متفاوتند.

هزینه‌های نرم افزاری در مهندسی متمرکز شده است و این بدان معناست که پروژه‌های نرم افزاری را نمی توان و نباید بصورت پروژه‌های تولیدی مدیریت نمود.



شکل ۱-۱ منحنی شکست سخت افزار

۲- نرم افزار فرسوده نمی شود

شکل ۱-۱ میزان شکست را به عنوان تابعی از زمان درخصوص سخت افزار نشان می دهد. این رابطه که اغلب «منحنی وارن شکل» نامیده می شود، نشانگر این است که سخت افزار در اوایل عمرش میزان عدم موفقیت نسبتاً بالایی دارد (این شکستها اغلب به طرح یا نقایص تولیدی نسبت داده می شوند). نقایص اصلاح شده و میزان شکست برای مدتی به سطح ثابتی می رسد (که البته خوشبختانه بسیار کم است) با گذشت زمان، سخت افزار شروع به فرسوده شدن کرده، میزان شکست کار دوباره افزایش می یابد.

نرم افزار در معرض آن عوامل محیطی که سخت افزار را خراب می کنند، نمی باشد. بنابراین، از نظر تئوری منحنی عدم موفقیت درمورد نرم افزار شکل یک منحنی ایده آل را می گیرد که در شکل ۲-۱ نشان داده شده است. مشکلات شناسایی نشده در اوایل کار برنامه باعث عدم موفقیت در کار می شود. این نقایص برطرف می شود (خوشبختانه بدون بوجود آوردن مشکلات دیگر) و منحنی همانگونه که در شکل آمده اوج می گیرد. منحنی ایده آل چیزی بالاتر از ساده سازی مدلهای واقعی ناموفق برای نرم افزار است (درمورد اطلاعات بیشتر به فصل ۸ مراجعه شود) در هر حال یک نکته کاملاً مشخص است. نرم افزار فرسوده نمی شود، اما کیفیت خود را از دست می دهد.

این تناقض را می توان به بهترین نحو با در نظر گرفتن منحنی واقعی که در شکل ۲-۱ آمده نشان داد. در طول حیات نرم افزار، این وسیله دستخوش تغییرات می شود (نگهداری)، با انجام تغییرات، احتمال دارد معایب جدیدی بوجود آید که باعث می شود منحنی میزان شکست همانگونه که در شکل ۲-۱ آمده، بالا رود. قبل از آنکه منحنی به وضعیت ثابت اصلی اش برگردد، تغییر دیگری مورد نیاز است که باعث می شود منحنی دوباره اوج بگیرد. به آرامی، سطح حداقل میزان شکست اوج می گیرد یعنی نرم افزار به خاطر تغییر، کیفیت خود را از دست می دهد.

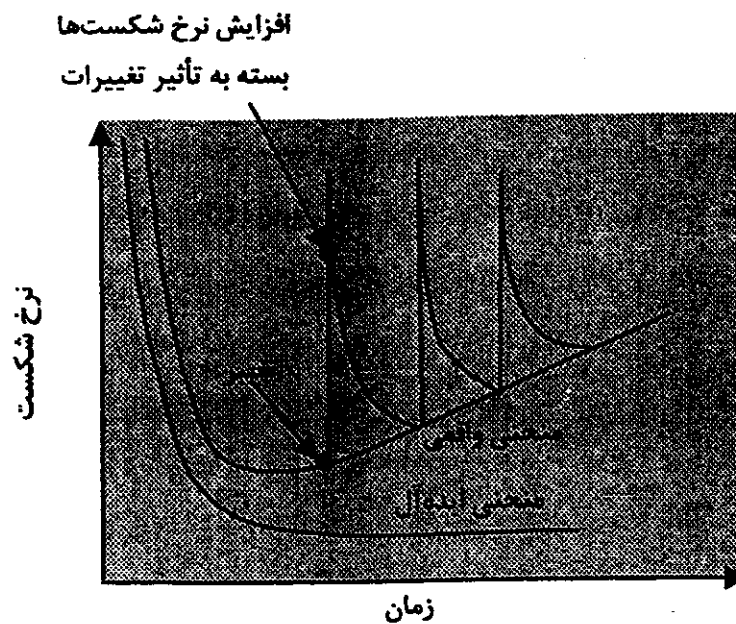
جنبه دیگر فرسودگی تفاوت بین سخت افزار و نرم افزار را وقتی نشان می دهد که یک جزء سخت افزاری فرسوده شده است و با یک قطعه یدکی جایگزین می شود. هر نقص نرم افزاری نشانگر یک خطا در طراحی یا فرایندی است که از طریق آن، طرح بصورت یک برنامه قابل اجرا در دستگاه در می آید. بنابراین نگهداری نرم افزار مستلزم پیچیدگیهای بیشتری نسبت به حفظ و نگهداری سخت افزار است.



نرم افزار کهنه نمی شود
اما زوال یافته و منسوخ
می گردد

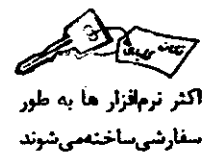


شیوه های مهندسی
نرم افزار سعی در کاهش
شیبهای منحنی واقعی
شکل ۲-۱ دارند



شکل ۱-۲ منحنی ایده آل و واقعی شکست برای نرم افزار

۲- گرچه صنعت به سمت مونتاژ اجزاء حرکت می کند ، اما نرم افزار همچنان سفارشی ساخته می شود.



وضعیتی را در نظر بگیرید که در آن سخت افزار کنترلی برای یک محصول کامپیوتری طراحی و تولید می شود. مهندس طراح یک طرح شماتیک ساده از مدار بندی رقمی رسم می نماید، قدری تحلیل اولیه انجام می دهد که از درستی کار مطمئن شود، سپس به سراغ قفسه ای می رود که کاتالوگ اجزاء رقمی وجود دارد. بعد از انتخاب هر جزء، از درون این قسمت خارج می شود.

با تکامل نظم و انضباط در طراحی، مجموعه ای از اجزای استاندارد طراحی بوجود می آید: پیچ های استاندارد و مدارهای منسجم متداول موجود در بازار تنها دو مورد از هزاران اجزاء استاندارد هستند که بوسیله مهندسان مکانیک و الکترونیک به عنوان وسایلی برای طراحی سیستم های جدید استفاده می شوند. بعضی قسمت ها بصورت مجدد قابل استفاده اند، بنابراین مهندس می تواند حواس خود را متمرکز بر عناصر واقعاً جدید و ابتکاری در طرح نماید، مانند بخش هایی از طرح که نمایانگر چیز جدیدی هستند. در جهان سخت افزار، استفاده مکرر از یک قطعه امری طبیعی در فرایند مهندسی است. در دنیای نرم افزار، کار طراحی تنها به این منظور شروع می شود که به یک مقیاس وسیع دست یابیم.

یک قطعه نرم افزاری باید طراحی و اجرا شود به گونه ای که در بسیاری از برنامه های مختلف دیگر استفاده شود. در دهه ۶۰، ما کتابخانه هایی علمی از زیر روالها درست نمودیم که در مجموعه گسترده ای از برنامه های کاربردی علمی و مهندسی قابلیت استفاده مجدد داشتند.



استفاده مجدد از
نرم افزار در فصل ۱۳
تشریح گردیده است.
مهندسی نرم افزار
مبتنی بر اجزاء نیز در
فصل ۲۷ ارائه گردید
است.

این کتابخانه های زیرروال، از الگوریتم هایی که قابل استفاده مجدد بودند استفاده می نمودند ولی دامنه کاربردی محدودی داشتند. امروزه ما دیدگاه خود را درخصوص استفاده مجدد، طوری بسط داده ایم که نه تنها شامل الگوریتمها بلکه شامل ساختمان داده ها نیز می شود. اجزاء مدرن قابل استفاده مجدد هم دارای داده می باشند و هم شیوه پردازش مخصوص آن داده ها را شامل هستند که مهندس نرم افزار را قادر می سازد، برنامه های کاربردی جدیدی را از روی قطعات قابل استفاده مجدد بسازد. برای مثال امروزه رابط های گرافیکی کاربر با استفاده از اجزای قابل استفاده مجدد ساخته می شوند که ایجاد پنجره های گرافیکی، منوهای باز شونده و یک سری مکانیزم های دوطرفه را میسر می سازد.

۲-۲-۱ کاربردهای نرم افزار

نرم افزار ممکن است در هر موقعیتی که مجموعه ای از مراحل رویه ای از پیش مشخص شده وجود دارد، بکار گرفته شود (مثل یک الگوریتم) (استثنائاتی نیز در این مورد وجود دارد مثل نرم افزار سیستم های خبره و نرم افزار شبکه عصبی). محتوای اطلاعات و میزان دقت و صحت و قطعیت آنها از عوامل مهم در تعیین ماهیت برنامه کاربردی نرم افزاری است. محتوای مطالب اشاره دارد به معنی و شکل اطلاعات ورودی و خروجی. مثلاً، بسیاری از برنامه های کاربردی تجاری از داده های ورودی دارای ساختار پیچیده استفاده کرده، (یک پایگاه داده ای) و گزارشات قالب بندی شده ای را ارائه می کنند. نرم افزاری که یک دستگاه خودکار را کنترل می کند (مثل یک کنترل عددی) قلم های داده ای مجزایی را با ساختار محدود پذیرفته و در یک توالی سریع فرمانهای هر دستگاه را صادر می کند.

قطعیت اطلاعات به قابلیت پیش بینی سفارش و زمانبندی اطلاعات اشاره دارد. یک برنامه تحلیل مهندسی، اطلاعاتی را می پذیرد که دترای یک سفارش از پیش تعیین شده بوده، الگوریتم های تحلیلی را بدون وقفه به اجرا درآورده و داده های منتج را بصورت گزارش یا قالب گرافیکی تولید می کند. چنین برنامه هایی درست و دقیق هستند. به عبارت دیگر یک سیستم عامل چندکاربره داده های ورودی را می پذیرد که دارای مطالب متنوع بوده و یک زمانبندی قراردادی دارند.

الگوریتم هایی را به اجرا درمی آورد که می توانند در شرایط خارجی تفسیر شوند و مطالب خروجی اش چیزهایی هستند که به عنوان تابعی از محیط و زمان متغیرند. برنامه های کاربردی که چنین خصوصیتی دارند قطعی نیستند.

لرانه طبقه بندی های کلی که درمورد برنامه های کاربردی نرم افزار با مفهوم باشند تاحدی سخت است. با افزایش پیچیدگی برنامه نرم افزار، عمل تفکیک کردن درست، از بین می رود. حوزه های نرم افزاری زیر نشانگر وسعت برنامه های کاربردی بالقوه هستند:

نرم افزار سیستم. نرم افزار سیستم مجموعه ای از برنامه ها است که برای سرویس دادن به سایر برنامه ها نوشته شده اند. بعضی از نرم افزارهای سیستم (مثل کامپایلرها، واژه پردازها و برنامه های سودمند

مدیریت فایل) ساختارهای اطلاعاتی پیچیده اما قطعی را پردازش می‌کنند. سایر برنامه‌های کاربردی سیستمها (مثل اجرای سیستم عامل، راه‌اندازها، پردازشگرهای ارتباطات از راه دور) تاحد زیادی داده‌های غیرقطعی را پردازش می‌کنند. در هر دو صورت حوزه نرم‌افزار سیستم با تأثیر شدیداً متقابل نرم‌افزار در مقابل سخت افزار کامپیوتری مواجه است. همینطور با استفاده گسترده توسط: چندکاربر، عملیات همروند که مستلزم زمانبندی، به اشتراک گذاری منابع و مدیریت فرآیند خبره است و نیز ساختار داده های پیچیده و رابط های چندگانه خارجی.

نرم‌افزار زمان واقعی^۱. نرم‌افزاری که نظارت، تحلیل و کنترل رویدادهای جهان واقعی را بر عهده دارد زمان واقعی نامیده می‌شود. عناصر نرم‌افزار زمان واقعی مشتمل بر اجزایی است که جمع آوری داده ها را بر عهده دارند که جمع آوری و قالب بندی اطلاعات از محیط خارجی را انجام می دهند، مولفه ای که تبدیل این اطلاعات را برعهده دارد، به گونه ای که برنامه کاربردی بتواند از آن استفاده نماید و مولفه ای که کنترل خروجی را عهده‌دار است و به محیط خارجی پاسخ می گوید و یک جزء کنترلی که همه قسمتهای دیگر را هماهنگ می‌کند، به طوریکه می‌توان واکنش فوری را حفظ نمود (که معمولاً از یک میلیونوم ثانیه تا یک ثانیه می‌باشد).

نرم‌افزار تجاری^۲. پردازش اطلاعات تجاری و بازرگانی یکی از بزرگترین حوزه‌های برنامه کاربردی نرم‌افزاری است.

سیستمهای گسسته (مثل پرداخت حقوق، حسابهای قابل دریافت / قابل پرداخت، موجودی و غیره) بصورت یک نرم‌افزار سیستم اطلاعات مدیریت تکمیل شده‌اند که به یک یا چند پایگاه داده‌ای بزرگ دسترسی دارد که حاوی اطلاعات تجاری هستند. برنامه‌های کاربردی این حوزه داده‌های موجود را به شکلی بازسازی می‌کند که عملیات تجاری یا تصمیم‌گیری‌های مدیریتی را تسهیل می‌کنند. علاوه بر برنامه کاربردی متعارف پردازش اطلاعات، برنامه‌های نرم‌افزاری تجاری حاوی محاسبه محاوره‌ای نیز هستند (مثل پردازش تراکنش نقطه فروش).

نرم‌افزار مهندسی و علمی^۳. این نرم‌افزار بوسیله یک سری الگوریتمهای پردازشگر عددی توصیف شده است. برنامه‌های کاربردی از ستاره‌شناسی تا آتشفشان شناسی، از تحلیل فشار اتومبیل تا دینامیک لرزه‌شناسی و از بیولوژی مولکولی تا تولید خودکار و اتوماتیک را دربر می‌گیرد. برنامه‌های کاربردی مدرن در حوزه عملی / مهندسی درحال دور شدن از الگوریتمهای عددی متعارف هستند. طراحی با کمک کامپیوتر، شبیه سازی سیستم و دیگر برنامه‌های کاربردی محاوره ای پلادرنگ شروع به کار نموده و حتی دارای مشخصه‌های نرم‌افزاری سیستم هستند.



ارجاع به وب

یکی از جامع ترین کتابخانه های قابل استفاده (آزاد برای عموم) در آدرس زیر قابل دسترسی است :
www.shareware.com

1. Real-time software

2. Business Software

3. Engineering and scientific software

نرم افزار جاسازی شده.^۱ محصولات هوشمند تقریباً در اختیار هر مشتری و در هر بازار صنعتی هستند. نرم افزار جاسازی شده در حافظه ROM (فقط خواندنی) جای گرفته و برای کنترل محصولات و سیستمهای مشتریان و بازارهای صنعتی استفاده می شوند. نرم افزار جاسازی شده می تواند کارهای بسیار محدود و سریع را انجام داده (مثل صفحه کلید کنترل در ماکروفر) یا کارهای مهم را انجام داده و میزان توانایی را تحت کنترل درآورد (مثل کارهای دیجیتالی در یک اتومبیل مثل کنترل سوخت، صفحه نمایش داشبورد، سیستم ترمز و غیره).

نرم افزار کامپیوتر شخصی.^۲ بازار نرم افزار کامپیوتر شخصی در طول دو دهه گذشته بسیار پررونق شده است. واژه پرداز، برنامه صفحه گسترده، گرافیکهای کامپیوتری، چند رسانه ای (مالتی مدیا)، تفریحات، مدیریت پایگاه دادهای، برنامههای کاربردی مالی تجاری و شخصی، شبکه خارجی و دسترسی به پایگاه دادهای تنها چند مورد از صدها برنامه کاربردی هستند.

نرم افزار برپایه وب.^۳ صفحات وب که توسط یک مرورگر بازایی شده اند، نرم افزاری هستند که دستورات قابل اجرا (مثل CGI، HTML، Perl یا Java) و دادهها (مثل فوق متن^۴ و یک سری قالبهای سمعی و بصری) را بکار می گیرند. در اصل، شبکه یک کامپیوتر حجیم است که تقریباً منابع نامحدود نرم افزاری را مهیا می کند که هر کسی می تواند با یک مودم به آن دسترسی پیدا کند.

نرم افزار هوش مصنوعی.^۵ نرم افزار هوش مصنوعی یا AI از الگوریتمهای غیر عددی برای حل مشکلات پیچیده ای استفاده می کند که در محاسبات یا تحلیلهای ساده قابل اصلاح نیستند. سیستمهای تخصصی که سیستمهای مبتنی بر دانش نیز نامیده می شوند، همراه یا شناسایی الگو (تصویر و صدا)، شبکههای عصبی مصنوعی، حل قضیه و بازیهای کامپیوتری نمونههایی از برنامههای کاربردی در این گروه می باشند.

۳-۱ نرم افزار: بحرانی در افق؟

بسیاری از ناظران کارهای صنعتی (از جمله خود نویسنده) مشکلات مربوط به نرم افزار را به عنوان «بحران» توصیف کرده اند. چند کتاب محدود [YOU98a]^۶ و [FLO97]^۷ و [GLA97]^۸ تأثیر بعضی از

1. Embedded Software

2. Personal computer software

3. Web-based software

4. Hypertext

5. artificial intelligence software

6. Yourdon, E

7. Flolowers, S

8. Glass, R.L.

نقل قول

آنچه بیشتر خبرگان معتقدند، معدوم شدن جهان بر اثر تصادف است. این بدن مناسبت که ما متخصصان کامپیوتر به وجود آورندگان آن

شکستهای نرم‌افزاری بارز دیگر را که در طول دهه گذشته رخ داده‌اند، بیان نموده‌اند. موفقیت‌های بزرگ صنعت نرم‌افزار بسیاری را به این پرسش واداشته که آیا هنوز عبارت «بحران نرم‌افزاری»^۱ مناسب نیست یا خیر. رابرت گلاس، نویسنده کتابهایی در زمینه عدم موفقیت نرم‌افزار، نماینده کسانی است که احساس‌اند تغییر کرده است. او بیان می‌کند [GLA98]^۲ که «من داستان شکستهای خود را بررسی می‌کنم و گزارش استثنایا را می‌بینم، شکستهای چشمگیری در اواسط بسیاری از موفقیتها، فنجانی که اکنون تقریباً پُر است.»

کلمه بحران در «فرهنگ وبستر» بعنوان نقطه عطفی در سیر هر چیز بیان شده، زمان مهم و قطعی، مرحله یا حادثه. از نظر کیفیت کلی نرم‌افزار و سرعتی که سیستم مبتنی بر کامپیوتر با آن کارکرد و محصولات ارائه شده، هیچگونه نقطه عطف یا زمان تصمیم‌گیری وجود ندارد. تنها یک تغییر بنیادی آهسته است که توسط تغییرات مهم فناوری در اصول مربوط به نرم‌افزار مشخص شده است.

هرکسی که کلمه Crisis را در فرهنگ لغت پیدا می‌کند معنی دیگری نیز برای آن می‌یابد «نقطه تحولی در سیر بیماری وقتی که معلوم می‌شود که آیا بیمار زنده می‌ماند یا می‌میرد». این تعریف در مورد ماهیت واقعی مشکلاتی که در سر راه توسعه نرم‌افزار وجود دارند، نکات کلیدی را در اختیار ما می‌گذارد.

واقعاً بجز " ناراحتی مزمن " ^۳ با چه چیز دیگری می‌توان توصیف بهتری داشت. کلمه affliction بصورت هر چیزی که باعث درد و یا ناراحتی می‌شود تعریف شده است. اما این معنی کلمه

Chronic یا مزمن است که در بحث ما نکته اصلی را تشکیل می‌دهد. «چیزی که مدتهای طولانی باقی مانده یا اغلب بروز کرده و بطور نامحدودی ادامه می‌یابد.» برای توصیف مشکلات بوجود آمده در کارهای نرم‌افزاری بهتر است از ناراحتی مزمن بجای بحران استفاده کنیم.

بدون توجه به آنچه که گفتیم مجموعه مشکلاتی که در توسعه نرم‌افزار کامپیوتری وجود دارد محدود به کار کرد درست نرم‌افزار نمی‌شود. بلکه این ناراحتی و نگرانی شامل مشکلات مربوط به چگونگی توسعه نرم‌افزار، چگونگی پشتیبانی از حجم روبه افزایش نرم‌افزار موجود و چگونگی انتظار ما در حفظ همراهی تقاضاهای روبه افزایش در مورد نرم‌افزارهای بیشتر است.

ما تا به امروز با این ناراحتی زندگی کرده‌ایم، در حقیقت صنایع علیرغم وجود آن موفق بوده‌اند و کارها بهتر هم خواهند شد اگر که بتوانیم یک معالجه را بصورت گسترده‌ای بکار گیریم.

1. software crisis

2. Glass, R.L.

۳. این لغت توسط پروفسور دانیل تیکرو (از دانشگاه میشیگان) در آوریل سال ۱۹۸۹ در یک ارائه گفتاری در ژنو، پیشنهاد شد.

۴-۱ افسانه های نرم افزار^۱

بسیاری از علل نقایص نرم افزاری را می توان در مسائلی دانست که در ابتدای بوجود آمدن نرم افزار شکل گرفته اند خلاف داستانهای قدیمی که اغلب درسهای عبرت آموز برای انسان فراهم می کردند که شنیدن آنها بسیار ارزشمند است، داستانهایی که در مورد نرم افزارها گفته می شد یکسری اطلاعات نادرست منتشر کرده و باعث سردرگمی می شد. داستانهای نرم افزاری دارای یک سری صفات هستند که آنها را غافلگیر کننده می سازد. مثلاً آنها به نظر جملات منطقی می آیند (که گاهی حاوی عناصر حقیقی هستند)، یک احساس درونی دارند و اغلب توسط افراد باتجربه ای که موضوع را می دانند، منتشر می شوند.

نقل قول

در غیاب استانداردهای معتاد، معنی نوبا چون نرم افزار متکی بر رسم و رسوم و سلیقه ما خواهد بود. نام دمارکو

افسانه های مدیریتی.

مدیرانی که دارای مسئولیت نرم افزاری هستند مثل مدیران اکثر رشته ها اغلب تحت فشارند تا بودجه را حفظ نموده، هرگونه خللی را در برنامه از بین برده و کیفیت را بهبود ببخشند. مثل کسی که در حال غرق شدن است و به هر چیزی چنگ می زند، یک مدیر نرم افزار نیز اغلب به این ایده در نرم افزار متوسل است البته اگر که این عقیده بتواند فشار را کم کند (حتی بطور موقت).

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

واقعیت: ممکن است کتاب استانداردها بخوبی وجود داشته باشد اما آیا استفاده شده؟ آیا متخصصین نرم افزار از وجود آن آگاهند؟ آیا این کتاب شیوه های مدرن مهندسی نرم افزار را ارائه می کند؟ آیا این کتاب کارآمد شده تا زمان تحویل را بهبود بخشد درعین حالیکه روی کیفیت نیز متمرکز است؛ در بسیاری از موارد، پاسخ به همه این سؤالات منفی است.

افسانه: ^۲ افراد من هنر توسعه ابزارهای نرم افزاری را دارند، ما برای آنها جدیدترین کامپیوترها را می خریم.

واقعیت: ^۲ ارائه و توسعه یک نرم افزار با کیفیت چیزی بیشتر از داشتن آخرین مدل یک مین فریم (کامپیوتر بزرگ اصلی)، ایستگاه کاری یا یک PC است. ابزارهای مهندسی نرم افزار با کمک کامپیوتر (Case) در رسیدن به کیفیت و بهره روری خوب بسیار مهمتر از سخت افزار هستند درحالیکه اکثر ارائه دهندگان نرم افزار هنوز از آنها بطور مؤثر استفاده نمی کنند.

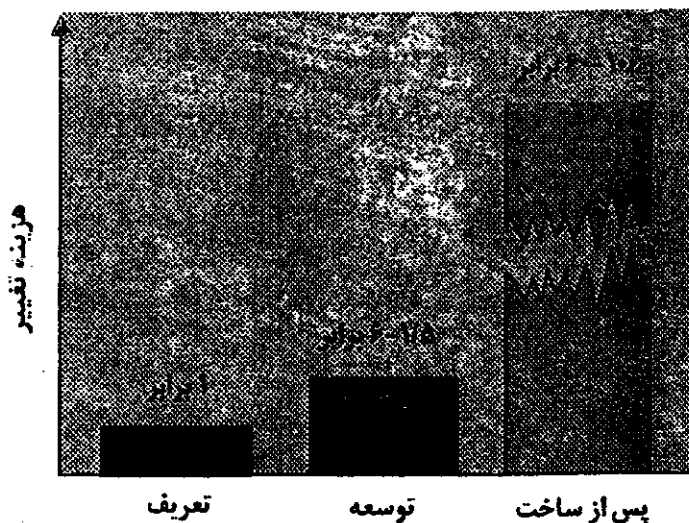
1. Management Myths

2. Myth

3. Reality

افسانه: اگر از جدول زمانبندی عقب بمانیم^۱، می‌توانیم به برنامه نویسان اضافه نموده و به جدول برسیم. (گاهی به آن مفهوم گروه مغولان می‌گویند).

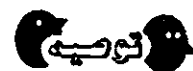
واقعیت: ارائه نرم‌افزار یک فرآیند مکانیکی مانند تولید و ساخت نیست. به بیان پروکر [BRO75] «... اضافه کردن افراد در اواخر کار پروژه کار را به تعویق می‌اندازد» ابتدا ممکن است این گفته حدسی به نظر برسد. با افزایش تعداد افراد جدید، افرادی که کار می‌کنند باید وقت خود را صرف آموزش تازه‌واردین کنند به همین خاطر میزان زمان صرف شده در مورد کارهای بهره‌وری کاهش می‌یابد. می‌توان به تعداد افراد افزود، اما تنها به شیوه‌ای برنامه ریزی شده و با هماهنگی کامل.



شکل ۳-۱ اثر تغییر

افسانه‌های مشتریان^۲

مشتری که تقاضای نرم‌افزار کامپیوتری می‌کند ممکن است شخصی در میز بفل دستی، یک گروه فنی در سالن پایین، بخش بازرگانی/فروش یا یک شرکت خارج از محیط کاری باشد که تحت قراردادی اقدام به درخواست نرم‌افزار نموده است. در بسیاری از موارد، مشتری افسانه‌هایی در مورد نرم‌افزار در سر دارد زیرا مدیر نرم‌افزار و متخصصان کار چندگانه برای اصلاح اطلاعات غلط انجام نمی‌دهند این داستانهای خوش خیالاته منجر به انتظارات غلطی شده (از جانب مشتری) و نهایتاً نارضایتی تولید کننده را به همراه دارد.



سخت کار کنید تا آنچه لازم است پیش از شروع بدانید. بدیهی است که دانستن تمام جزئیات برای شما ممکن نخواهد بود. اما هرچه بیشتر بدانید، با مخاطرات کمتری مواجه خواهید شد.

۱. اصطلاح "منابع خارجی" به قراردادهایی اطلاق می‌شود که کار توسعه نرم‌افزار را به شخص ثالثی (معمولاً یک حرفه‌ای مشاور در ساخت نرم‌افزار های سفارشی خاص) می‌سپارد.

افسانه: وضعیت کلی اهداف مورد نظر برای شروع برنامه نویسی کافی است - می‌توانیم جزئیات را بعداً درست کنیم.

واقعیت: تعاریف ناقص کل به جزء دلیل اصلی شکست کارهای نرم‌افزاری است. یک توصیف دقیق و رسمی از دامنه اطلاعات، عملکرد، رفتار، نمایش، رابطه‌ها، محدودیتهای طراحی و معیارهای اعتبار ضروری است این مشخصه‌ها را تنها بعد از ارتباط کامل بین مشتری و ارائه دهنده، نرم‌افزار، می‌توان تعیین نمود.

افسانه: موارد مورد نیاز و شرایط پروژه دائماً تغییر می‌کنند اما با تغییر می‌توان به راحتی سازگار شد زیرا نرم‌افزار خاصیت تغییر پذیری دارد.

واقعیت: درست است که شرایط نرم‌افزاری تغییر می‌کنند، اما تأثیر تغییر نسبت به زمانی که این تغییر در آن رخ داده، فرق می‌کند. شکل ۱-۳، تأثیر تغییر را نشان می‌دهد. اگر جداً به تعریف داده شده توجه کنیم، می‌توان با تقاضاهای اولیه مبنی بر تغییر، تطابق پیدا کرد. مشتری می‌تواند تقاضاها را مرور نموده و اصلاحاتی را که تأثیر نسبتاً کمی روی هزینه دارد، توصیه کند وقتی تغییرات در طول طراحی نرم‌افزار ارائه می‌شوند، تأثیر هزینه به سرعت افزایش می‌یابد. یک سری منابع برای کار تهیه شده و چارچوب کاری، طراحی شده است، هر گونه تغییر ممکن است باعث آشوب و هرج و مرج شود که نیازمند منابع اضافی و اصلاح بیشتر طرح است، یعنی افزایش هزینه، تغییر عملکرد، کار، رابطه‌ها یا میان مشخصه‌ها در طول اجرای برنامه (کد و آزمون) تأثیر سریعی روی قیمت و هزینه می‌گذارند. تغییر در صورتیکه بعد از آماده شدن نرم‌افزار درخواست شود، چیزی بیشتر از یک سفارش بوده و ممکن است هزینه بیشتری نسبت به ارائه همان تغییر در زمان زودتر داشته باشد.

رؤیاهای یک متخصص^۱

افسانه‌هایی که یک سری متخصص هنوز هم به آنها اعتقاد دارند بعد از گذشت ۵۰ سال از زمان فرهنگ برنامه نویسی، پرورش یافته‌اند. در طول روزهای اولیه با گرفتن نرم‌افزار، برنامه نویسی به صورت یک فرم از هنر تلقی می‌شد. شیوه‌ها و نگرشهای قدیمی به سختی از بین می‌روند.

افسانه: وقتی ما برنامه را می‌نویسیم و به راه می‌اندازیم، کارمان تمام می‌شود.

واقعیت: زمانی کسی گفت: که «هر چه زودتر کار نوشتن کد را آغاز کنید، دیرتر کارتان تمام می‌شود.» اطلاعات صنعتی [PUT97]، [JON91]، [LIE80] نشان می‌دهد که بین ۶۰ و ۸۰ درصد از کل تلاشهای صورت گرفته در مورد نرم‌افزار تازه بعد از آنکه اولین بار به دست مشتری می‌رسند، هزینه می‌شوند.

افسانه: تا وقتی که برنامه را اجرا کنیم، هیچ شیوه‌ای برای دسترسی به کیفیت آن ندارم.



مدیریت و کنترل تغییرات، به تفصیل در فصل ۹ آورده شده است. { کتاب هدایت تغییر مترجم نیز قدری راه گشا خواهد بود }



هرگاه گمان کردید که زمان کافی برای دیسپلین‌های مهندسی نرم‌افزار در اختیار ندارید، از خود بپرسید: "آیا زمان کافی برای انجام مجدد این پروژه از ابتدا خواهی داشت؟"

واقعیت: یکی از موثرترین مکانیزمهای تضمین کیفیت نرم‌افزار از زمان شروع پروژه بکار گرفته می‌شود و آن باز بینی رسمی^۱ آن نام دارد. باز بینی‌های نرم‌افزاری (که در فصل ۸ آمده) فیلترهای کیفیتی هستند که مؤثرتر از آزمون برای یافتن گروه‌های معینی از نقایص نرم‌افزاری می‌باشند.

افسانه: تنها جزء قابل ارائه برای یک پروژه موفق برنامه کاری است.

واقعیت: برنامه کاری تنها بخشی از "پیکربندی نرم‌افزار"^۱ است که شامل عناصر بسیاری می‌باشند. مستندسازی بنیان یک مهندسی موفقیت آمیز بوده و مهمتر اینکه راهنمایی برای حمایت نرم‌افزاری مهابی سازد.

بسیاری از متخصصان نرم‌افزار غلط بودن مسائل تشریح شده، فوق پی برده‌اند. در کمال تأسف، شیوه‌ها و روشهای سنتی مدیریت و روشهای فنی ضعیفتر را، حتی وقتی واقعیت روش بهتر را نشان می‌دهد، دنبال می‌کنند. شناخت واقعیت‌های نرم‌افزاری اولین مرحله در رسیدن به ترکیب مناسب راه حل‌های عملی برای ارائه نرم‌افزار است.

۵-۱ خلاصه

نرم‌افزار عنصر اصلی در تکامل سیستمهای مبتنی بر کامپیوتر و محصولات آن است. در طول ۵۰ سال گذشته نرم‌افزار از یک وسیله رفع مشکل تخصصی و یک ابزار تحلیل اطلاعاتی به یک صنعت مستقل تبدیل شده است. اما تاریخچه و فرهنگ برنامه نویسی اولیه یک سری مسائل را بوجود آورده که امروزه نیز وجود دارند. نرم‌افزار در تکامل مستمر سیستمهای کامپیوتری یک عامل محدود کننده محسوب می‌شود. نرم‌افزار متشکل از برنامه‌ها، داده‌ها و مستندات است. هر کدام از آنها قسمتی از پیکربندی فرایند مهندسی نرم‌افزار خواهد بود. هدف از مهندسی نرم‌افزار تأمین چارچوبی برای ساخت نرم‌افزاری با کیفیت بالاتر می‌باشد.

مسایل و نکاتی برای تفکر و تعمق بیشتر

۱-۱ نرم افزار در بسیاری از سیستمها و محصولات کامپیوتری عنصر اصلی متمایز کننده است. نمونه هایی از دو یا سه محصول و حداقل یک سیستم را ارائه دهید که در آن نرم افزار و نه سخت افزار عنصر اصلی متمایز کننده باشد.

۲-۱ در دهه ۵۰ و ۶۰ برنامه سازی کامپیوتر یک نوع هنر بود که طی دوره کار آموزی در محیطی فرا گرفته می شد، روزهای نخستین، امروزه در ارائه نرم افزار چه تأثیری گذاشته است ؟

۳-۱ بسیاری از متخصصین در مورد " عصر اطلاعات " بحث کرده اند، چند مثال ارائه دهید (چه مثبت و چه منفی) که اثر نرم افزار را روی جامعه نشان دهد. یکی از مراجع قبل از سال ۱۹۹۰ را در بخش ۱-۱ باز بینی نموده، نشان دهید که پیش بینی های نویسنده در کجا درست بوده و در کجا غلط بوده است.

۴-۱ یک برنامه کاربردی خاص را انتخاب نموده نشان دهید:

الف) برنامه کاربردی نرم افزاری، از نظر رده بندی در کجا قرار می گیرد؟

ب) محتوای داده ای مربوط به برنامه کاربردی چیست؟

ج) و اطلاعات دقیق برنامه کاربردی چه می باشد.

۵-۱ با گسترش و عمومیت هر چه بیشتر نرم افزار مخاطرات آن برای مردم (به جهت برنامه های اشتباه) به نگرانی جمعی تبدیل می شود. یک نمونه، بنیانی دیگر بجز Y2K ارائه دهید که در آن شکست برنامه کامپیوتری باعث ضرر و زیان شدید می شود.

۶-۱ در اینترنت به دقت گروه خبری comp.risks را مطالعه کنید و خلاصه ای از خطرات که عموم مردم با آن مواجه بوده اند و اخیراً مورد بحث فرا گرفته اند، ارائه دهید، منبع دیگر

Software engineering Notes از انتشارات ACM

۷-۱ مقاله ای بنویسید و در آن پیشرفتهای اخیر در یک جنبه از حوزه برنامه های کاربردی نرم افزاری را بطور خلاصه ذکر کنید، انتخابهای بالقوه شامل: برنامه های کاربردی پیشرفته بر پایه وب، شبکه های عصبی مصنوعی، رابطه های پیشرفته انسانی و عوامل هوشمند است.

۸-۱ نکاتی که در قالب «فلسفه» در بخش ۴-۱ ارائه شد در طی سالهای گذشته به کندی رنگ باخته اند و چیزهای دیگری جای آنها را گرفته اند. سعی کنید یکی دو مورد جدید از این موارد را به هر رده بیافزائید.

فهرست منابع و مراجع

- [BR075] Brooks, F., *The Mythical Man-Month*, Addison-Wesley, 1975.
- [DEJ98] De Jager, P. et al., *Countdown Y2K: Business Survival Planning for the Year 2000*, Wiley, 1998.
- [DEM95] DeMarco, T., *Why Does Software Cost So Much?* Dorset House, 1995, p. 9.
- [FEI83] Feigenbaum, E.A. and P. McCorduck, *The Fifth Generation*, Addison-Wesley, 1983.
- [FL097] Flowers, S., *Software Failure, Management Failure-Amazing Stories and cautionary Tales*, Wiley, 1997.
- [GLA97] Glass, R.L., *Software Runaways*, Prentice-Hall, 1997.
- [GLA98] Glass, R.L., "Is There Really a Software Crisis?" *IEEE Software*, vol. 15, no. 1, January 1998, pp. 104-105.
- [HAM93] Hammer, M., and J. Champy, *Reengineering the Corporation*, HarperCollins Publishers, 1993.
- [JON91] Jones, c., *Applied Software Measurement*, McGraw-Hill, 1991.
- [KAR99] Karlson, E. and J. Kolber, *A Basic Introduction to Y2K: How the Year 2000 Computer Crisis Affects You*, Next Era Publications, 1999.
- [LEV95] Levy, S., "The Luddites Are Back," *Newsweek*, July 12, 1995, p. 55.
- [LEV99] Levy, S., "The New Digital Galaxy," *Newsweek*, May 31, 1999, p. 57.
- [LIE80] Lientz, B. and E. Swanson, *Software Maintenance Management*, Addison Wesley, 1980.
- [NAI82] Naisbitt, J., *Megatrends*, Warner Books, 1982.
- [NOR98] Norman, D., *The Invisible Computer*, MIT Press, 1998.
- [OSB79] Osborne, A., *Running Wild-The Next Industrial Revolution*, Osborne/McGraw-Hill, 1979.
- [PUT97] Putnam, L. and W. Myers, *Industrial Strength Software*, IEEE Computer Society Press, 1997.
- [STO89] Stoll, c., *The Cuckoo's Egg*, Doubleday, 1989.
- [TOF80] Tomer, A., *The Third Wave*, Morrow, 1980.
- [TOF90] Tomer, A., *Powershift*, Bantam Publishers, 1990.
- [YOU92] Yourdon, E., *The Decline and Fall of the American Programmer*, Yourdon Press, 1992.
- [YOU96] Yourdon, E., *The Rise and Resurrection of the American Programmer*, Yourdon Press, 1996.
- [YOU98a] Yourdon, E., *Death March Projects*, Prentice-Hall, 1998.
- [YOU98b] Yourdon, E. and J. Yourdon, *Time Bomb 2000*, Prentice-Hall, 1998.

خواندنیهای دیگر و منابع اطلاعاتی

Literally thousands of books are written about computer software. The vast majority discuss programming languages or software applications, but a few discuss software itself. Pressman and Herron (*Software Shock*, Dorset House, 1991) presented an early discussion (directed at the layperson) of software and the way professionals build it.

Negroponte's (*Being Digital*, Alfred A. Knopf, 1995) best-selling book provides a view of computing and its overall impact in the twenty-first century. Books by Norman [NOR98] and Bergman (*Information Appliances and Beyond*, Academic Press/Morgan Kaufmann, 2000) suggest that the widespread impact of the PC will decline as

information appliances and pervasive computing connect everyone in the industrialized world and almost every "appliance" that they own to a new Internet infrastructure.

Minasi (*The Software Conspiracy: Why Software Companies Put out Faulty Products, How They Can Hurt You, and What You Can Do*, McGraw-Hill, 2000) argues that the "modern scourge" of software bugs can be eliminated and suggests ways to accomplish this. DeMarco (*Why Does Software Cost So Much?* Dorset House, 1995) has produced a collection of amusing and insightful essays on software and the process through which it is developed.

A wide variety of information sources on software-related topics and management is available on the Internet. An up-to-date list of World Wide Web references that are relevant to software can be found at the SEPA Web site:

<http://www.mhhe.com/engcs/compsci/pressman/resources/product.mhtml>

این کتاب تنها به خاطر حل مشکل دانشجویان پیام نور تبدیل به پی دی اف شد. همین جا از ناشر و نویسنده و تمام کسانی که با افزایش قیمت کتاب مارا مجبور به این کار کردند و یا متحمل ضرر شدند عذرخواهی می کنم.
گروهی از دانشجویان مهندسی کامپیوتر مرکز تهران

