



# مهندسی نرم افزار

با رویکرد علمی و کاربردی

مؤلف:

دکتر محمدعلی ترکمانی

سرشناسه : ترکمانی ، محمدعلی ، ۱۳۵۴-

عنوان و نام پدیدآور : مهندسی نرم افزار با رویکرد علمی و کاربردی / مولف محمدعلی ترکمانی .

مشخصات نشر : مشهد: ارسسطو، ۱۳۹۵.

مشخصات ظاهري : ۲۳۶ ص.

شابک : ۹۷۸-۶۰۰-۴۳۲-۰۶۳-۴

وضعیت فهرست نویسی: فیبا

یادداشت: کتابنامه: ص. ۲۳۶.

موضوع: نرم افزار—مهندسی

موضوع: Software engineering

رده‌بندی کنگره : QA۷۶/۷۵۸/۴۹۵ م ۱۳۹۵

رده‌بندی دیوبی: ۱/۰۵

شماره کتابشناسی ملی: ۴۳۴۶۴۹۱

نام کتاب : مهندسی نرم افزار با رویکرد علمی و کاربردی

مؤلف : دکتر محمدعلی ترکمانی

ناشر : ارسسطو (با همکاری سامانه اطلاع‌رسانی چاپ و نشر ایران )

صفحه‌آرایی ، تنظیم و طرح جلد : محمدعلی ترکمانی و علی بیات

تیراژ: ۱۰۰۰ جلد

نوبت چاپ : سوم - ۱۳۹۸

چاپ : مدیران

قیمت : ۵۵۰۰ تومان

شابک : ۹۷۸-۶۰۰-۴۳۲-۰۶۳-۴

تلفن‌های مرکز پخش : ۰۹۱۷۷۱۶۴۹۴۰ - ۰۹۱۴۶ - ۰۵۱۱

این اثر مشمول قانون حمایت از مولفان و مصنفان و هنرمندان است. هر کس تمام یا قسمتی از

این اثر را بدون اجازه مولف نشر یا پخش یا عرضه کند، مورد پیگرد قانونی قرار خواهد گرفت.

## فهرست مطالب

<b>فصل اول: مهندسی نرم افزار چیست؟.....</b>	<b>۱۷</b>
۱-۱- بحران نرم افزار .....	۱۷
۱-۲- تعریف مهندسی نرم افزار .....	۱۸
۱-۳- اهداف مهندسی نرم افزار .....	۱۸
۱-۴- اصول مهندسی نرم افزار .....	۱۸
۱-۴-۱- فرآیند .....	۱۸
۱-۴-۱-۱- فعالیتهای چتری مهندسی نرم افزار .....	۲۰
۱-۴-۲- متداولوژی .....	۲۰
۱-۴-۳- ابزارهای مهندسی نرم افزار به کمک کامپیوتر .....	۲۰
۱-۴-۵- عوامل پدیدآورنده پیچیدگی نرم افزار .....	۲۰
۱-۶- سؤالات تشریحی .....	۲۲
<b>فصل دوم: فرآیند تولید نرم افزار.....</b>	<b>۲۳</b>
۲-۱- انواع مدل های فرآیند ارائه شده .....	۲۳
۲-۱-۱- مدل ترتیبی خطی .....	۲۳
۲-۱-۲- مدل آبشاری .....	۲۳
۲-۱-۳- مدل نمونه سازی .....	۲۵
۲-۲-۱- مدل توسعه سریع کاربرد (RAD) .....	۲۶
۲-۲-۲- مدل های تکاملی فرآیند توسعه نرم افزار .....	۲۸
۲-۲-۳- مدل افزایشی (تکرار و توسعه تدریجی) .....	۲۸
۲-۲-۴- مدل حلزونی .....	۲۹

۳۰	.....Win Win ۲-۲-۳
۳۱	.....مدل مونتاژ مؤلفه‌ها ۲-۲-۴
۳۱	.....مدل توسعه هم‌روند ۲-۲-۵
۳۳	.....روش‌های رسمی توسعه نرم‌افزار ۲-۲-۶
۳۴	.....مشخصات فرایندهای توسعه موفق ۲-۳
۳۴	.....مدل‌سازی تصویری نرم‌افزار ۲-۳-۱
۳۵	.....UML چیست؟ ۲-۳-۱-۱
۳۶	.....دیدهای مختلف UML ۲-۳-۱-۲
۳۶	.....نمودارهای UML ۲-۳-۱-۳
۳۷	.....سوالات تشریحی ۲-۴
۱۲۶	<b>فصل سوم: متدولوزی RUP</b>
۱۲۶	.....فرایند RUP ۳-۱
۱۲۶	.....چرخه حیات توسعه نرم‌افزار در RUP ۳-۲
۱۳۰	.....نظم‌های RUP ۳-۲-۱
۱۳۰	.....فرآورده‌ها ۳-۲-۲
۱۳۱	.....سوالات تشریحی ۳-۳
۳۹	<b>فصل چهارم: اصول شی‌گرایی و کلاس‌ها</b>
۳۹	.....مقدمه ۴-۱
۳۹	.....مفاهیم اساسی ۴-۲
۴۰	.....نمایش کلاس در UML ۴-۳
۴۰	.....مکانیزم ارتباط بین اشیاء ۴-۴
۴۰	.....اصل شی‌گرایی ۴-۵

۴۱	..... ۱-۵-۴-تجربید یا انتزاع
۴۲	..... ۱-۵-۴-نقش تجربید در کنترل پیچیدگی
۴۲	..... ۲-۵-۴-محصورسازی یا پنهانسازی جزئیات یا کپسوله سازی
۴۳	..... ۱-۵-۴-نقش محصورسازی در کنترل پیچیدگی
۴۳	..... ۳-۵-۴- واحد بندی
۴۳	..... ۱-۵-۴-ویژگی‌های واحد بندی
۴۴	..... ۲-۳-۵-۴-اصل مازولاریتی
۴۵	..... ۳-۳-۵-۴-نقش واحد بندی و تکنیک تجزیه در کنترل پیچیدگی
۴۵	..... ۴-۳-۵-۴-مزایای طراحی پیمانه‌ای کارآمد
۴۵	..... ۴-۵-۴-سلسله‌مراتب
۴۶	..... ۱-۴-۵-۴-نقش سلسله‌مراتب در کنترل پیچیدگی
۴۷	..... ۶-۴-سوالات تشریحی
۴۹	..... <b>فصل پنجم: مدل‌سازی موارد کاربری</b>
۴۹	..... ۱-۵-مفاهیم اساسی مدل مورد کاربری
۵۲	..... ۲-۵-سازمان‌دهی موارد کاربری
۵۲	..... ۱-۲-۵-بسته‌بندی
۵۳	..... ۲-۲-۵-رابطه وراثت
۵۳	..... ۳-۲-۵-رابطه دربراشت
۵۴	..... ۴-۲-۵-رابطه گسترش دادن
۵۴	..... ۳-۵-نحوه ایجاد مدل موارد کاربری
۵۶	..... ۴-۵-سوالات تشریحی
۵۷	..... <b>فصل ششم: مدل‌سازی کلاس‌ها</b>
۵۷	..... ۱-۶-مقدمه

۵۷	۶-۲-۱- رابطه انجمنی
۵۷	۶-۲-۲- سلسله مراتب Part-of یا رابطه تجمعی
۵۸	۶-۲-۳- رابطه عام/خاص یا رابطه وراثت
۵۹	۶-۳-۱- مدل تحلیل
۶۱	۶-۳-۲- مدل طراحی
۶۲	۶-۳-۳- مدل کلیشه در UML
۶۳	۶-۴- روش‌های شناسایی کلاس‌ها
۶۴	۶-۵- انواع رابطه بین کلاس
۶۶	۶-۶- سؤالات تشریحی
۷۵	۷-۱- ساختار پویای سیستم
۷۶	۷-۲- نمودار ترتیبی
۷۷	۷-۳- نمایش شرط و حلقه در نمودار ترتیبی
۷۹	۷-۴- نمودار حالت
۸۱	۷-۵- مفاهیم اساسی نمودار حالت
۸۲	۷-۶- مفاهیم نمودار فعالیت
۸۵	۷-۷- موارد استفاده نمودار فعالیت
۸۵	۷-۸- مفاهیم نمودار فعالیت
۸۷	۷-۹- مثال: نمودار فعالیت ثبت تصادفات
۸۷	۷-۱۰- سؤالات تشریحی

<b>فصل هشتم: مدل‌سازی مؤلفه و استقرار</b>	<b>۸۹</b>
۸۹ ..... بسته‌ها	۸-۱
۸۹ ..... وارد کردن بسته‌ها	۸-۱-۱
۹۰ ..... قابلیت دستیابی بسته	۸-۱-۲
۹۰ ..... رابطه وراثت در بسته‌ها	۸-۱-۳
۹۱ ..... رابطه وابستگی بین بسته‌ها	۸-۱-۴
۹۲ ..... زیرسیستم	۸-۲
۹۳ ..... چه وقت زیرسیستم‌ها استفاده می‌شوند؟	۸-۲-۱
۹۳ ..... مدل‌سازی مؤلفه‌ها	۸-۳
۹۴ ..... تفاوت‌های شیء‌گرایی و توسعه بر مبنای مؤلفه	۸-۳-۱
۹۴ ..... ویژگی‌های مشترک مؤلفه و شیء	۸-۳-۲
۹۵ ..... تفاوت‌های بین مؤلفه و کلاس	۸-۳-۳
۹۵ ..... نقش واسط در مؤلفه‌ها	۸-۳-۴
۹۷ ..... انواع مؤلفه‌های در UML	۸-۳-۵
۹۸ ..... مدل‌سازی استقرار	۸-۴
۱۰۰ ..... سوالات تشریحی	۸-۵
<b>فصل نهم: ویژگی‌های کیفی در مهندسی نرم‌افزار</b>	<b>۱۰۱</b>
۱۰۱ ..... خصوصیات کیفی	۹-۱
۱۰۲ ..... خصوصیات کیفی سیستمی	۹-۱-۱
۱۰۲ ..... قابلیت اصلاح (Modifiability)	۹-۱-۱-۱
۱۰۲ ..... قابلیت آزمایش (Testability)	۹-۱-۱-۲
۱۰۲ ..... قابلیت استفاده (Usability)	۹-۱-۱-۳

۱۰۲	۹-۱-۱-۴-کارایی (Performance)
۱۰۳	۹-۱-۱-۵-قابلیت اعتماد (Reliability)
۱۰۳	۹-۱-۱-۶-امنیت (Security)
۱۰۳	۹-۱-۱-۷-قابلیت نگهداری (Maintainability)
۱۰۴	۹-۱-۱-۸-حملپذیری (Portability)
۱۰۴	۹-۱-۱-۹-قابلیت دسترسی (Availability)
۱۰۴	۹-۱-۱-۱۰-قابلیت استفاده مجدد (Reuseability)
۱۰۴	۹-۱-۱-۱۱-قابلیت کار متقابل (Interoperability)
۱۰۴	۹-۱-۲-خصوصیات کیفی حرفه
۱۰۵	۹-۱-۳-خصوصیات کیفی معماری
۱۰۶	۹-۲-سوالات تشریحی
<b>۱۰۷</b>	<b>فصل دهم: مفاهیم طراحی</b>
۱۰۷	۱۰-۱-طراحی در حیطه مهندسی نرم افزار
۱۰۷	۱۰-۲-انواع طراحی
۱۰۸	۱۰-۳-فرایند طراحی
۱۰۸	۱۰-۴-خصوصیات طراحی خوب
۱۰۹	۱۰-۵-مفاهیم طراحی
۱۰۹	۱۰-۵-۱-انتزاع یا تجرید (Abstraction)
۱۰۹	۱۰-۵-۲-الگوهای (patterns)
۱۰۹	۱۰-۵-۳-جداسازی دغدغه‌ها (Separation of concerns)
۱۰۹	۱۰-۵-۴-پیمانه بندی (Modularity)
۱۱۰	۱۰-۵-۵-پنهان سازی اطلاعات (Information Hiding)
۱۱۰	۱۰-۵-۶-استقلال عملیاتی (Functional Independence)

۱۱۰ .....	(Refinement) پالایش ۷-۵-۱۰
۱۱۰ .....	(Aspects) جنبه‌ها ۸-۵-۱۰
۱۱۱ .....	(Refactoring) بازآرایی ۹-۵-۱۰
۱۱۱ .....	(Design classes) انواع کلاس‌های طراحی ۱۰-۵-۱۰
۱۱۲ .....	ویژگیهای کلاس‌های طراحی خوش فرم ۱۱-۵-۱۰
۱۱۲ .....	(DESIGN MODEL) مدل طراحی ۶-۱۰-۱۰
۱۱۳ .....	سوالات تشریحی ۷-۱۰-۱۰
<b>۱۱۶ .....</b>	<b>فصل یازدهم: طراحی معماری نرم افزار</b>
۱۱۶ .....	معماری نرم افزار ۱-۱۱
۱۱۶ .....	توصیف‌های معماری ۲-۱۱
۱۱۷ .....	تصمیم‌گیری‌های معماری ۳-۱۱
۱۱۷ .....	سبک‌های معماری ۴-۱۱
۱۱۷ .....	معرفی برخی از سبک‌های معماری ۱-۱۱
۱۱۷ .....	معماری‌های داده محور ۱-۱۱-۴-۱
۱۱۸ .....	سبک شیء‌گرا ۲-۱۱-۴-۱
۱۱۸ .....	معماری لایه‌ای ۳-۱۱-۴-۱
۱۱۹ .....	معماری MVC برای برنامه‌های کاربردی تحت وب ۴-۱-۱۱
۱۲۰ .....	تاکتیک‌ها، راهبردها و الگوهای معماری ۵-۱۱
۱۲۱ .....	طراحی معماری ۶-۱۱
۱۲۲ .....	روش تحلیل توازن‌های معماری (ATAM) ۷-۱۱
۱۲۴ .....	روش تحلیل هزینه فایده (CBAM) ۸-۱۱
۱۲۴ .....	سوالات تشریحی ۹-۱۱

<b>فصل دوازدهم: روش‌های چابک توسعه نرم‌افزار.....</b>	<b>۱۳۲</b>
۱۲-۱- مقدمه.....	۱۳۲
۱۲-۲- تفاوت متدولوژی‌های سبک وزن و سنگین وزن .....	۱۳۲
۱۲-۳- بیانیه چابک.....	۱۳۶
۱۲-۴- متدولوژی XP.....	۱۳۶
۱۲-۴-۱- ویژگی‌ها.....	۱۳۶
۱۲-۴-۲- اصول و ارزش‌ها .....	۱۳۷
۱۲-۴-۳- مدل فرایندی XP .....	۱۳۸
۱۲-۴-۴- نقش‌ها و مسئولیت‌ها .....	۱۳۹
۱۲-۴-۵- فراورده‌های XP .....	۱۴۰
۱۲-۴-۶- فعالیت‌های XP .....	۱۴۱
۱۲-۵- معرفی SCRUM .....	۱۴۳
۱۲-۵-۱- جلسات اسکرام .....	۱۴۶
۱۲-۵-۲- نقش‌ها و مسئولیت‌ها .....	۱۴۷
۱۲-۶- مزایا و معایب متدولوژی‌های چابک.....	۱۴۸
۱۲-۷- سؤالات تشریحی .....	۱۴۹
<b>فصل سیزدهم: راهبردهای آزمون نرم‌افزار.....</b>	<b>۱۵۱</b>
۱۳-۱- مقدمه.....	۱۵۱
۱۳-۲- ویژگی‌های راهبردهای آزمون نرم‌افزار.....	۱۵۱
۱۳-۳- سطوح راهبرد آزمون .....	۱۵۲
۱۳-۳-۱- وارسی و اعتبارستنجدی (V&V).....	۱۵۲
۱۳-۴- سازمان‌دهی برای آزمون نرم‌افزار .....	۱۵۳

۱۵۳	..... ۱۳-۵ راهبردی برای آزمایش نرم افزار
۱۵۴	..... ۱۳-۶ اصول آزمایش
۱۵۵	..... ۱۳-۷ راهبردهای آزمون برای نرم افزارهای سنتی
۱۵۵	..... ۱۳-۷-۱ آزمایش واحد
۱۵۶	..... ۱۳-۷-۲ آزمایش یکپارچه سازی یا انسجام
۱۵۶	..... ۱۳-۷-۲-۱ انواع آزمون های انسجام
۱۵۶	..... ۱ آزمایش رگرسیون
۱۵۷	..... ۲ آزمایش دود
۱۵۷	..... ۸ آزمون واحدها در حیطه‌ی OO (شیء‌گرا)
۱۵۸	..... ۹ راهبردهای آزمون برای برنامه‌های تحت وب
۱۵۸	..... ۱۰ آزمون اعتبار سنجی
۱۵۸	..... ۱۰-۱ ملاک‌های آزمون اعتبار سنجی
۱۵۹	..... ۱۰-۱-۱ بازبینی پیکربندی
۱۵۹	..... ۱۱ آزمایش‌های آلفا و بتا و پذیرش
۱۶۰	..... ۱۲ آزمون سیستم
۱۶۰	..... ۱۲-۱ آزمون ترمیم (Recovery Testing)
۱۶۰	..... ۱۲-۲ آزمون امنیت (Security Testing)
۱۶۱	..... ۱۲-۳ آزمون فشار (Stress Testing)
۱۶۱	..... ۱۲-۴ آزمون کارایی (performance Testing)
۱۶۱	..... ۱۲-۵ آزمون استقرار (Deployment Testing)
۱۶۲	..... ۱۳ دیدگاه درونی و بیرونی نسبت به آزمون
۱۶۲	..... ۱۴ آزمایش جعبه سفید

۱۶۳ .....	۱۴-۱-آزمایش مسیر پایه
۱۶۳ .....	۱۴-۲-آزمایش حلقه
۱۶۴ .....	۱۵-آزمایش جعبه سیاه
۱۶۴ .....	۱۵-۱-تحلیل مقادیر مرزی
۱۶۵ .....	۱۶-سوالات تشریحی
<b>۱۶۷ .....</b>	<b>فصل چهاردهم: مدیریت پیکربندی نرم افزار</b>
۱۶۷ .....	۱-۱۴-مدیریت پیکربندی نرم افزار
۱۶۷ .....	۲-چالش های مدیریت پیکربندی
۱۶۸ .....	۳-منشاء تغییرات
۱۶۸ .....	۴-خط مبنا
۱۶۹ .....	۵-اقلام پیکربندی
۱۷۰ .....	۶-فرایند SCM
۱۷۰ .....	۱-۱۴-۶-شناسایی اشیای پیکربندی نرم افزار
۱۷۰ .....	۲-۱۴-۶-کنترل نسخه
۱۷۰ .....	۳-۱۴-۶-کنترل تغییر
۱۷۲ .....	۴-۱۴-۶-ممیزی پیکربندی
۱۷۳ .....	۵-۱۴-۶-تهیه گزارش وضعیت پیکربندی
۱۷۳ .....	۷-۱۴-۶-سوالات تشریحی
<b>۱۷۵ .....</b>	<b>فصل پانزدهم: برنامه ریزی پروژه های نرم افزاری</b>
۱۷۵ .....	۱-۱۵-۱-مقدمه
۱۷۵ .....	۲-۱۵-۲-برآوردها
۱۷۵ .....	۱-۱۵-۲-۱-گامهای برآوردها

۱۷۶ .....	۱۵-۳-برآوردهای پروژه نرم افزاری
۱۷۷ .....	۱۵-۴-روش های تجزیه
۱۷۷ .....	۱۵-۴-۱- تعیین اندازه نرم افزار
۱۷۷ .....	۱۵-۴-۱-۲- معیارهای سنجش مبتنی بر اندازه
۱۷۸ .....	۱۵-۴-۱-۳- معیارهای سنجش مبتنی برتابع(عملکرد).
۱۸۰ .....	۱۵-۴-۲- برآوردهای مبتنی برمسئله.
۱۸۱ .....	۱۵-۵-برآوردهای پروژه های شیء گرا
۱۸۱ .....	۱۵-۶-برآوردهای توسعه چابک
۱۸۱ .....	۱۷-۷-برآوردهای برنامه کاربردی وب
۱۸۲ .....	۱۷-۸-تصمیم گیری ساخت یا خرید
۱۸۲ .....	۱۷-۹-سوالات تشریحی
<b>۱۸۳ .....</b>	<b>فصل شانزدهم: مدلسازی UML با ویژوال پارادایم</b>
۱۸۳ .....	۱۶-۱-معرفی ویژوال پارادایم
۱۸۳ .....	۱۶-۲-نصب نرم افزار
۱۸۴ .....	۱۶-۳-نمای کلی رابط کاربر
۱۸۵ .....	۱۶-۴-باز کردن / بستن قابها
۱۸۵ .....	۱۶-۵-ایجاد پروژه جدید
۱۸۸ .....	۱۶-۶-ذخیره سازی پروژه
۱۸۸ .....	۱۶-۷-تنظیمات فونت و رنگ
۱۸۹ .....	۱۶-۸- تنظیم جعبه ابزار DIAGRAMS
۱۹۱ .....	۱۶-۹-سوئیچ کردن بین دیاگرام های مختلف یک پروژه
۱۹۴ .....	۱۶-۱۰-رسم نمودار موارد کاربری

۱۹۴	..... روشن اول	۱۰-۱-۱۶
۱۹۷	..... روشن دوم	۲-۱۰-۱-۱۶
۱۹۸	..... ایجاد نمودار کلاس	۱۱-۱۶-۱
۱۹۸	..... نحوه ساخت کلاس	۱-۱۱-۱۶-۱
۱۹۹	..... ساخت رابطه همکاری	۲-۱۱-۱۶
۲۰۰	..... تجمعیع	۳-۱۱-۱۶
۲۰۱	..... تعیین نوع رابطه بین کلاس‌ها	۴-۱۱-۱۶
۲۰۳	..... ایجاد رابطه وراثت	۵-۱۱-۱۶
۲۰۴	..... رابطه وابستگی بین دو کلاس	۶-۱۱-۱۶
۲۰۴	..... ایجاد صفت کلاس	۷-۱۱-۱۶
۲۰۵	..... ایجاد عملکردهای کلاس	۸-۱۱-۱۶
۲۰۶	..... مرتب‌سازی اعضای کلاس	۹-۱۱-۱۶
۲۰۷	..... ترسیم نمودار فعالیت	۱۲-۱۶
۲۰۷	..... ایجاد Swimlane	۱-۱۲-۱۶
۲۰۸	..... الحق پارتیشن به Swimlane	۲-۱۲-۱۶
۲۰۹	..... ایجاد گره‌های اولیه	۳-۱۲-۱۶
۲۰۹	..... ایجاد عملیات (کنش)	۴-۱۲-۱۶
۲۱۲	..... تفاوت Activity و Action	۵-۱۲-۱۶
۲۱۲	..... توسعه نمودار توالی	۱۳-۱۶
۲۱۵	..... ایجاد نمودار بسته	۱۴-۱۶
۲۱۵	..... تعیین کلیشه	۱-۱۴-۱۶
۲۱۷	..... ایجاد نمودار مؤلفه	۱۵-۱۶

۲۱۹ .....	۱۶-۱۵-۱- تعیین کلیشه برای مؤلفه
۲۲۰ .....	۱۶-۱۵-۲- تعریف واسطه‌ها
۲۲۰ .....	۱۶-۱۵-۲-۱- تعریف واسط ارائه
۲۲۱ .....	۱۶-۱۵-۲-۲- تعریف واسط موردنیاز
۲۲۱ .....	۱۶-۱۵-۳- ایجاد وابستگی بین مؤلفه‌ها
۲۲۲ .....	۱۶-۱۶-۱- ایجاد نمودار استقرار
۲۲۲ .....	۱۶-۱۶-۱- ایجاد گره‌ها و ترسیم ارتباطات بین گره‌ها
۲۲۴ .....	۱۶-۱۶-۲- ایجاد نمودار حالت
۲۲۵ .....	۱۶-۱۶-۲-۱- ایجاد حالت‌ها و انتقال‌ها
۲۲۷ .....	۱۶-۱۶-۲-۲- اضافه کردن ناحیه به حالت
۲۲۸ .....	۱۶-۱۶-۲-۳- مدل‌سازی خصوصیات انتقال
۲۳۰ .....	۱۶-۱۶-۳- نمایش محتوای حالت در نمودار حالت
۲۳۳ .....	۱۶-۱۷-۱- ترسیم زیرسیستم
۲۳۷ .....	۱۶-۱۷-۱- تمرین
۲۳۷ .....	ضمیمه: نمونه سوالات امتحان پایانی
۲۳۸ .....	منابع:

## مقدمه:

از بدو پیدایش نرم افزار تاکنون، تحويل به موقع نرم افزار، کاهش هزینه ها، مدیریت پیکربندی و اعمال تغییرات و تولید یک محصول با کیفیت و رضایت مشتری همواره از دغدغه های توسعه دهنده کان سیستم های نرم افزاری بوده است. سوابق اجرای پروژه های نرم افزاری نشان می دهد که بسیاری از پروژه ها با شکست مواجه شده اند. با توجه به مواردی که ذکر شد، در نیمه دوم قرن بیستم بحث بحران نرم افزار مطرح گردید و به دنبال آن مهندسی نرم افزار به عنوان یک راه حل برای غلبه بر این بحران مطرح شد. مهندسی نرم افزار یک درس تخصصی در رشته های مهندسی کامپیوتر - نرم افزار، مهندسی فناوری اطلاعات (تمامی گرایش ها) و مهندسی ICT گرایش دیتا است. مباحث مهندسی نرم افزار در مقطع کارشناسی دانشگاه های مختلف با عنوانی نظیر تحلیل و طراحی سیستم ها، مهندسی نرم افزار و تحلیل و طراحی شیء گرا ارائه می گردد. این کتاب سعی دارد مباحث موردنیاز دانشجویان کارشناسی را با زبانی ساده و به صورت کاربردی بیان نماید. در پایان هر فصل تعدادی سؤال تشریحی آورده شده است تا دانشجویان گرامی بهتر بتوانند خود را برای آزمون های پیش رو آماده نمایند. همچنین در این کتاب آموزش نرم افزار ویژوال پارادایم که یک نرم افزار کامل برای مدل سازی پروژه های نرم افزاری است، ارائه گردیده است. امید است این اثر مورد توجه همکاران و دانشجویان گرامی قرار گیرد. از استاید و دانشجویان گرامی تقاضا دارم دیدگاه های خود را از طریق [ایمیل](mailto:m.a.torkamani@gmail.com) با اینجانب در میان بگذارند تا انشالله در ویرایش های بعدی اشکالات یا کاستی های احتمالی کتاب مورد تجدیدنظر قرار گیرد. در پایان وظیفه خود می دانم از خدمات آقای مهندس علی بیات به خاطر طراحی جلد کتاب و همچنین از مدیریت انتشارات ارسسطو و سامانه اطلاع رسانی چاپ و نشر ایران، جناب آقای حسین قنبری، تشکر و قدردانی نمایم.

محمدعلی ترکمانی

تابستان ۱۳۹۵

# فصل اول

## مهندسی نرم افزار چیست؟

### ۱- بحران نرم افزار

در نیمه دوم قرن بیستم، پیشرفت شگرف سخت افزار و ارزان شدن آن موجب افزایش تقاضا برای سیستم‌های نرم افزاری شده است. این موضوع در کنار ضعف روش‌های تولید نرم افزار و ناتوانی این روش‌ها در کنترل پیچیدگی نرم افزار بحران نرم افزار<sup>۱</sup> را بوجود آورد. علاوه بر این بحران (معضلات ناشی از ضعف روش‌های توسعه نرم افزار در کنترل پیچیدگی نرم افزار) عبارتند از:

- عدم بهره گیری کامل از قدرت سخت افزار
- ناتوانی روش‌های تولید نرم افزار در پاسخگوئی به افزایش تقاضا
- هزینه‌های هنگفت تولید نرم افزار
- عدم تحويل به موقع نرم افزار
- عدم تامین نیازمندیهای کاربر
- کیفیت پایین و نامطمئن
- سختی نگهداری بعلت کیفیت پایین طراحی
- برای مقابله با این بحران مهندسی نرم افزار مطرح شد.

## ۱-۲- تعریف مهندسی نرم افزار

از دیدگاه آقای Pressman به مجموعه‌ای از فن‌آوری‌ها، روش‌ها و ابزارهای مبتنی بر اصول مهندسی که در توسعه نرم افزار استفاده می‌شود، مهندسی نرم افزار گفته می‌شود. مهندسی نرم افزار عبارت است از به کارگیری یک روش سیستماتیک، منظم و قابل اندازه‌گیری برای تولید و توسعه، عملیاتی کردن و نگهداری نرم افزار و همچنین به کارگیری اصول مهندسی در تولید نرم افزار به منظور تولید یک نرم افزار باکیفیت.

## ۱-۳- اهداف مهندسی نرم افزار

- دستیابی به نیازمندی‌های کیفی و وظیفه‌مندی نرم افزار: نیازمندی‌های وظیفه‌ای کارهایی هستند که نرم افزار باید انجام دهد. مانند ارائه گزارش‌های مختلف، فرم‌های ورود و ویرایش اطلاعات و غیره. نیازهای کیفی مانند قابلیت دسترسی، امنیت، قابلیت توسعه، قابلیت استفاده مجدد و قابلیت اطمینان.
- رضایت کاربر و ذینفعان
- کاهش هزینه‌های تولید و نگهداری نرم افزار
- تحويل بهموقع نرم افزار
- استفاده از مؤلفه‌های استاندارد
- استفاده مجدد

## ۱-۴- اصول مهندسی نرم افزار

شکل ۱-۱ تکنولوژی مهندسی نرم افزار را نشان می‌دهد که از سه اصل یا لایه تشکیل شده است.

### ۱-۴-۱- فرایند

وقتی کار می‌کنید تا یک سیستم یا محصول را سازید حتماً باید یک سری مراحل قابل پیش‌بینی را چک کنید. یک نقشه راه که در ایجاد محصول باکیفیت شما را یاری می‌کند. این نقشه فرایند نرم افزار نام دارد. فرایند مجموعه‌ای از فعالیت و کوششی است در جهت رسیدن به هدفی گسترده.

در حیطه مهندسی نرم افزار فرایند یک روش انطباق پذیر است که تیم نرم افزار به کمک آن می‌توانند مجموعه‌ای مناسب از کنش‌ها و وظایف کاری را برگزینند.



شکل ۱-۱: لایه‌های مهندسی نرم افزار

یک کنش شامل مجموعه‌ای از وظایف است که یک محصول کاری عمدۀ را تولید می‌کند. چارچوب فرایند کلی برای مهندسی نرم افزار شامل پنج فعالیت می‌شود: ۱- ارتباطات ۲- برنامه‌ریزی ۳- مدل‌سازی ۴- ساخت ۵- استقرار. یک فرآیند تولید نرم افزار چهار نقش اساسی دارد:

- ۱) ارائه مدلی برای فرایند تولید (مدل فرایند)<sup>۱</sup> یا پارادایم مهندسی نرم افزار که گام‌های لازم برای توسعه نرم افزار (ترتیب فعالیت‌هایی که باید انجام شود) و چگونگی انتقال از یک گام به گام دیگر را بیان می‌کند.
- ۲) بیان اینکه چه فرآورده‌هایی باید تولید شوند و در چه زمانی.
- ۳) تعیین روش اداره وظایف توسعه‌دهندگان منفرد و تیمی، نقش‌های موردنیاز در پروژه (تحلیل گر سیستم، معمار،...) و انتساب این نقش‌ها به اعضای تیم.
- ۴) فراهم نمودن معیارهایی<sup>۲</sup> برای اندازه‌گیری کیفیت محصولات پروژه و روند پیشرفت فعالیت‌های آن.

۱ process model

۲ Paradigm

۳ Artifacts

۴ Software Metrics

### ۱-۴-۱-۱- فعالیتهای چتری مهندسی نرم افزار

فعالیتهایی چتری مهندسی مدیریتی هستند که در سرتاسر پروژه انجام میشوند و به تیم نرم افزاری کمک می کند تا پیشرفت ، کیفیت ، تغییر و ریسک را کنترل کند. فعالیتهای چتری عبارتند از ۱- کنترل و پیگیری پروژه های نرم افزاری ۲- مدیریت ریسک ۳- تضمین کیفیت نرم افزار ۴- بازبینی های فنی ۵- اندازه گیری ۶- مدیریت پیکربندی نرم افزار ۷- مدیریت قابلیت استفاده ۸- تهیه و تولید محصول کاری مجدد.

### ۱-۴-۲- متدولوژی

منظور از متدها، روش است مانند متدهای تحلیل (مثل تحلیل شیء‌گرا). متداولوژی مجموعه‌ای از متدهایی است که در تمام چرخه حیات سیستم نرم افزاری اعمال شده و بر یک نوع نگرش کلی درباره جهان نرم افزار متکی می‌باشد. به نظر بسیاری از متخصصین به کارگیری یک متداولوژی مدون در تولید نرم افزار می‌تواند تا اندازه قابل توجهی مشکلات بیان شده را برطرف سازد.

### ۱-۴-۳- ابزارهای مهندسی نرم افزار به کمک کامپیوتر

در شکل ۱-۱ منظور از ابزارهای خودکار سازی، ابزارهای مهندسی نرم افزار به کمک کامپیوتر (CASE) است. وجود ابزار اتوماتیک برای کمک به تولید و اجرای مدل‌های مبتنی بر متداولوژی نیز از اهمیت زیادی برخوردار است. این ابزارها در فازهای مختلف به کار می‌روند. به عنوان نمونه‌ای از این ابزارها می‌توان رشنال رز و ویژوال پارادایم را نام برد.

## ۱-۵- عوامل پدیدآورنده پیچیدگی نرم افزار

معمولًا سیستم‌های نرم افزاری بزرگ حاوی عناصری است که پیچیدگی آن‌ها اجتناب‌پذیر است:  
۱- پیچیدگی حوزه مسئله: