

۲۳۴۲۹۲۶

کدنویسی تمیز

اثر رابرت سی مارتین

ترجمه

محمد رضا شیروانی

www.ketab.ir

شنوان و نام پدیدآور	: کدنویسی تمیز/ اثر رابرت سی مارتین ... و دیگران : مترجم محمدرضا شیروانی.
مشخصات نشر	: تهران: آروین نگار، ۱۴۰۱.
مشخصات ظاهری	: ۴۲۸ ص: مصور، نمودار.
شابک	: ۹۷۸-۶۲۲-۵۷۵۶-۲۱-۲
یادداشت	: Clean code : a handbook of agile software craftsmanship. عنوان اصلی:
عنوان دیگر	: کدنویسی تمیز clean code : راهنمای توسعه نرم افزار به روش چابک.
موضوع	: تولید نرم افزار به روش اگیل Agile software development نرم افزار — اطمینان پذیری Computer software -- Reliability
شناسه افزوده	: مارتین، رابرت سی، C، Martin, Robert
شناسه افزوده	: شیروانی، محمدرضا، ۱۳۴۷ - مترجم
رده بندی کنگره	: QA۷۶۷۶
رده بندی دیویی	: ۱/۰۰۵
شماره کتابشناسی ملی	: ۹۰۳۰۸۹۶
اطلاعات رکورد کتابشناسی	: فیپا

این اثر مشمول قانون حمایت مؤلفان و مصنفان و هنرمندان مصوب ۱۳۴۸ است. هرکس تمام یا قسمتی از این اثر را بدون اجازه (ناشر) نشر یا پخش یا عرضه کند مورد پیگرد قانونی قرار خواهد گرفت.



تلفن: ۶۶۴۱۸۵۱۲

همراه: ۰۹۳۹۱۲۶۱۴۱۹

مرکز پخش: میدان انقلاب، خیابان انقلاب نرسیده به ۲ فروردین، پلاک ۱۳۱۴، طبقه سوم، واحد ۱۱

عنوان کتاب کدنویسی تمیز
مؤلف رابرت سی مارتین
مترجم محمدرضا شیروانی
ناشر انتشارات آروین نگار
مدیر فنی و ناظر چاپ محرمعلی قاصدی
طراح جلد علی ذوالفقاری
نوبت و سال چاپ اول / ۱۴۰۱
تیراژ ۱۰۰ نسخه
قیمت ۲۲۰۰۰۰ تومان

مرکز پخش: فروشگاه اینترنتی کتاب آرتین www.artinbook.ir

تلفن: ۰۹۱۲۴۱۶۹۰۹ - ۶۶۴۸۱۸۷۰

پست الکترونیکی arvinnegarpub@gmail.com

کلیه حقوق این کتاب برای آروین نگار محفوظ است.

فهرست مطالب

پیش‌گفتار.....	۵
فصل اول: کدنویسی تمیز.....	۱۳
فصل دوم: نام‌های معنی‌دار.....	۳۳
فصل سوم: توابع.....	۵۱
فصل چهارم: کامنت‌ها.....	۷۷
فصل پنجم: قالب‌بندی.....	۱۰۳
فصل ششم: اشیاء و ساختمان داده.....	۱۲۳
فصل هفتم: مدیریت خطا.....	۱۳۵
فصل هشتم: مرزها.....	۱۴۹
فصل نهم: تست‌های واحد.....	۱۵۹
فصل دهم: کلاس‌ها.....	۱۷۵
فصل یازدهم: سیستم‌ها.....	۱۹۵
فصل دوازدهم: ضرورت.....	۲۱۷
فصل سیزدهم: هم‌زمانی.....	۲۲۵
فصل چهاردهم: پالایش متوالی.....	۲۴۵
فصل پانزدهم: اجزای داخلی JUnit.....	۳۱۵
فصل شانزدهم: بازتولید SerialData.....	۳۳۳
فصل هفدهم: ناپسامانی‌ها و اکتشافات.....	۳۵۵
پیوست الف: هم‌زمانی II.....	۳۹۳

پیش گفتار

یکی از آبنیات‌های موردعلاقه ما در دانمارک "Ga-Jol" است که به‌خاطر شیرینی زیاد آن، مکمل خوبی برای آب‌وهوای مرطوب و اغلب سرد ما است. بخشی از جذابیت "Ga-Jol" برای ما دانمارکی‌ها ضرب‌المثل‌های جالب و طنزآمیزی است که بر روی درب جعبه‌های آن چاپ می‌شود. امروز صبح یک بسته دوتایی از این خوراکی لذیذ را خریدم و متوجه شدم که این ضرب‌المثل قدیمی دانمارکی روی بسته آن نوشته شده است:

Ærlighed i små ting er ikke nogen lille ting.

"صداقت در چیزهای کوچک، چیز کمی نیست."

این یک مثال بسیار خوب مطابق با چیزی بود که من می‌خواستم در اینجا مطرح کنم. چیزهای کوچک مهم هستند. این کتاب به موضوعات کوچکی می‌پردازد که ارزش بالایی دارند.

معمار معروف "لودویگ میس ون در روه" گفته است: "عظمت خداوند را می‌توان در جزئیات مشاهده کرد." این نقل‌قول استدلال‌های معاصر در مورد نقش معماری در توسعه نرم‌افزار و به‌ویژه در دنیای توسعه چابک را یادآور می‌شود. من و "باب" گاهی اوقات خودمان را به‌شدت درگیر این بحث می‌کردیم. بله، "میس ون در روه" به کاربرد وسایل ماندگار یک ساختمان که زیربنای معماری‌های بزرگ هستند توجه ویژه‌ای داشت. از سوی دیگر، او برای هر خانه‌ای که طراحی می‌کرد، شخصاً دستگیره‌های درب را انتخاب می‌کرد. چرا؟ چون چیزهای کوچک مهم هستند.

"باب" و من در بحث‌های روزمره خود در مورد TDD متوجه شدیم که هر دوی ما موافق این هستیم که معماری نرم‌افزار جایگاه مهمی در فرایند توسعه دارد، اگرچه که ممکن است دیدگاه‌های متفاوتی درباره معنای دقیق معماری داشته باشیم. بااین‌حال، چنین حاشیه‌هایی زیاد مهم نیستند، زیرا ما هر دو اعتقاد داریم که همه متخصصان، زمانی را برای تفکر و برنامه‌ریزی در ابتدای هر پروژه اختصاص می‌دهند. اواخر دهه ۱۹۹۰ مفاهیم طراحی که تنها با تست‌ها و کدها هدایت می‌شدند، مدت‌هاست که از بین رفته‌اند. با این‌حال توجه به جزئیات حتی از هر دیدگاه دیگری پایه و اساس حرفه‌ای‌گری محسوب می‌شود. اولاً، از طریق تمرین در کارهای کوچک است که متخصصان مهارت و اعتمادبه‌نفس شروع کارهای بزرگ را کسب می‌کنند. دوم این که کوچک‌ترین ساختار شلخته‌ای مانند دری که به‌خوبی بسته نمی‌شود یا کاشی‌هایی که به‌صورت کج‌روی زمین قرار گرفته‌اند، یا حتی یک میز کشیف، جذابیت کل مجموعه را کاملاً از بین می‌برد. این دقیقاً همان چیزی است که به کدهای تمیز مربوط می‌شود.

معماری در حال حاضر هم یک استعاره خوب در توسعه نرم‌افزار و به‌ویژه برای بعضی از نرم‌افزارهایی که محصول اولیه را ارائه می‌دهند، به‌حساب می‌آید. به همان ترتیبی که یک معمار ساختمان اولیه را تحویل می‌دهد. در دنیای scrum و چابک ارائه سریع محصول به بازار از اصلی‌ترین

مراحل محسوب می‌شود. ما می‌خواهیم اجرای فاکتوری (کارخانه‌ای)^۱ با نهایت سرعت در جهت تولید نرم‌افزار صورت گیرد.

این موارد فاکتورهای انسانی هستند: تفکر و احساس کدنویسانی که با وجود انباشته شدن محصولات قدیمی، محصولات جدید خلق می‌کنند. استعاره کارخانه در این‌چنین تفکری خود را به‌خوبی نشان می‌دهد. صنعت خودروسازی ژاپن در دنیای خط مونتاژ، بیشتر از Scrum ها الهام می‌گیرد.

با این‌وجود، حتی در صنعت خودروسازی، بخش عمده کار در قسمت تولید نیست، بلکه در تعمیر و نگهداری نهفته است. در نرم‌افزار، نزدیک به ۸۰ درصد کاری که انجام می‌دهیم، به‌طور عجیبی به نگهداری مربوط می‌شود: "تعمیرات". به‌جای پذیرش مدل غربی یعنی تمرکز بر تولید نرم‌افزار خوب، باید بیشتر شبیه به تعمیرکاران خانگی در صنعت ساختمان یا مکانیک خودرو در صنعت خودروسازی فکر کنیم. مدیریت ژاپنی در این مورد چه می‌گوید؟

در حدود سال ۱۹۵۱، یک رویکرد کیفی به نام تعمیر و نگهداری بهره‌وری کل (TPM) در ژاپن مطرح شد. تمرکز آن به‌جای تولید بر روی تعمیر و نگهداری بود. یکی از ارکان اصلی TPM مجموعه‌ای از اصول به‌اصطلاح S5 بود. S5 مجموعه‌ای از اصول انضباطی است. در اینجا من از اصطلاح "انضباط" جهت هدف آموزشی خود استفاده می‌کنم. این اصول S5 در واقع پایه‌های حرفه‌ای بودن^۲ را تشکیل می‌دهند. معنا و مفهوم حرفه‌ای بودن یکی دیگر از اصطلاحات جوامع غربی در حوزه نرم‌افزار محسوب می‌شود. این اصول یک گزینه اختیاری نیستند. همان‌طور که عمو "باب" در مقدمه کتاب خودش می‌گوید: "اجرا و فعالیت یک نرم‌افزار خوب به‌چنین نظمی نیاز دارد: تمرکز، حضور ذهن و تفکر." همیشه موضوع اصلی فقط انجام کار نیست، بلکه فشار آوردن به تجهیزات کارخانه‌ای جهت تولید در سرعت بهینه است. فلسفه S5 شامل این مفاهیم است:

- Seiri، یا سازمان (به انگلیسی "sort" گفته می‌شود): درک امور با استفاده از روش‌های نام‌گذاری مناسب امری مهم و ضروری به‌حساب می‌آید. آیا شما فکر می‌کنید که نام‌گذاری شناسه‌ها مهم نیستند؟ برای پاسخ‌دادن به این پرسش فصل‌های بعدی را حتماً بخوانید.
- Seiton، یا نظم (به انگلیسی "systematize" گفته می‌شود): یک ضرب‌المثل قدیمی آمریکایی وجود دارد که می‌گوید: "یک مکان برای همه چیز و همه چیز در مکان خود". یک قطعه کد باید در جایی باشد که انتظار دارید به‌راحتی آن را پیدا کنید. اگر این‌طور نیست، باید دوباره کدنویسی کنید تا در جای مناسبش قرار گیرد.

1 Factory running

2 Lean