اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد مؤلفه‌های نرم‌افزاری: روش‌ها، کاربردها و چالش‌ها

بهاره بافنده مایوان[۱]، عباس رسول‌زادگان[۲*]

اطلاعات مقاله

دریافت مقاله: ۱۳۹۳/۱۲/۰۳
پذیرش مقاله:۱۳۹۴/۰۱/۲۶

واژگان کلیدی:
قابلیت استفاده‌ی مجدد از نرم-افزار،
توسعه نرم‌افزار مبتنی بر مؤلفه،
اندازه‌گیری کمّی،
روش‌های ارزیابی جعبه سیاه،
روش‌های ارزیابی جعبه سفید.

چکیده

هدف اصلی استفاده‌ی مجدد از مؤلفه‌های نرم‌افزاری، تولید نرم‌افزارهای جدید با استفاده از مؤلفه‌های موجود به جای تولید آن‌ها از ابتدا است. توسعه‌ی مبتنی بر مؤلفه، یکی از روش‌های مطرح توسعه نرم‌افزار می‌باشد. امروزه در گرایش‌های مختلف مهندسی، ترکیب مؤلفه‌های قابل استفاده‌ی مجدد و باکیفیت از اهمیت بسزایی برخوردار است اما در مهندسی نرم‌افزار، استفاده‌ی مجدد از مؤلفه‌ها هنوز با چالش‌های متعددی مواجه است. به‌منظور افزایش کارآمدی و سرعت بخشیدن به فرآیند استفاده‌ی مجدد در توسعه‌ی نرم‌افزار، اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد مؤلفه‌ها امری ضروری می‌باشد. در این راستا مدل‌ها و سنجه‌های مختلفی ارائه شده‌اند. با توجه به تعدّد روش‌های موجود، ضرورت ارائه‌ی یک مرور جامع، کامل و منصفانه در این زمینه، به شدّت احساس می‌گردد. در این مقاله، با انجام یک مرور نظام‌مند و با ارائه‌ی یک دسته‌بندی جدید، روش‌های موجود به دقت مورد بررسی و مطالعه قرار گرفته‌اند و به تحلیل نقاط قوّت، محدودیت‌ها و کاربردهای آن‌ها پرداخته شده‌است. در نهایت، بر مبنای نتایج مطالعات و تحلیل-های صورت پذیرفته، برخی از چالش‌ها و مسائل باز مرتبط معرفی شده‌اند.

۱- مقدمه

امروزه توسعه نرم‌افزار مبتنی بر مؤلفه[۱] به عنوان یک چارچوب کارآمد در توسعه نرم‌افزار در بسیاری از کاربردها مورد استفاده قرار می‌گیرد [۱، ۲، ۷، ۱۸]. توسعه مبتنی بر مؤلفه، تأکید بر طراحی و ساخت سیستم‌های نرم‌افزاری با استفاده از مؤلفه‌های قابل استفاده مجدد[۲] دارد. اگر در فرآیند تولید و توسعه یک سیستم نرم‌افزاری، تمام اجزای آن از ابتدا تولید شوند، فرآیند توسعه، بسیار پرهزینه خواهدبود اما در صورتیکه از مؤلفه‌های از پیش‌ساخته شده و آماده به جای تولید مجدد هریک از اجزای سیستم استفاده شود، هزینه‌های تولید با کاهش محسوسی همراه می‌گردد [۵، ۱۱]. از این‌رو یکی از مفاهیم پایه در مهندسی نرم‌افزار، قابلیت استفاده مجدد[۳] مؤلفه نرم‌افزاری است. از مزایای مهم این مفهوم هزینه پایین، تحویل به‌موقع، انعطاف‌پذیری، کیفیت، نگهداری کارآمد و مقیاس‌پذیری می‌باشد [۳]. در جدول (۱) تعدادی از مهمترین مزایای توسعه مبتنی بر مؤلفه از دیدگاه محققین مختلف عنوان شده‌است.

استفاده مجدد از مؤلفه‌های باکیفیت یکی از مهم‌ترین مسائل موجود در توسعه نرم‌افزار مبتنی بر مؤلفه است. با توجه به اهمیت و به‌کارگیری مفهوم توسعه مبتنی بر مؤلفه در کاربردهای مختلف، اندازه‌گیری کمّی[۴]

۱. دانشجوی دکترا، دانشکده مهندسی، دانشگاه فردوسی مشهد
* rasoolzadegan@um.ac.ir
۲. استادیار، دانشکده مهندسی، دانشگاه فردوسی مشهد

جدول (۱): مزایای توسعه‌ی مبتنی بر مؤلفه از دیدگاه محققین مختلف

پارامترها / نویسندگان	مقرون به‌صرفگی و کاهش هزینه‌ها	افزایش قابلیت اطمینان	افزایش قابلیت نگهداری	مقبولیت توامان در صنعت و در دانشگاه	افزایش کارایی	افزایش قابلیت استفاده‌ی مجدد
Washizaki et al. [۵]	*	*				
Lingyun et al. [۷]	*	*	*			
Singh et al. [۶]				*		
Hu et al. [۸]	*				*	*
Zhang et al. [۹]						*

قابلیت استفاده مجدد مؤلفه، امری ضروری می‌باشد، به‌طوریکه کاربر قادر باشد از بین چندین مؤلفه با عملکرد یکسان، مناسب‌ترین مؤلفه را از نظر قابلیت استفاده مجدد انتخاب کند. لذا باید معیارهای مناسبی برای اندازه‌گیری کمّی سطح قابلیت استفاده‌ی مجدد مؤلفه‌ها تعریف شوند [۱۲-۱۳]. با توجه به این‌که در این راستا روش‌های مختلفی ارائه شده‌است [۵-۳۴] ارائه‌ی مروری جامع در این زمینه لازم می‌باشد. تاکنون مرورهایی در این حوزه انجام شده‌است [۳۵-۴۲]، اما هیچ‌کدام از آن‌ها از جامعیت لازم برخوردار نیستند. در این مقاله، با انجام یک مرور نظام‌مند[۵] که سه ویژگی بارز آن، جامعیت، کمال و عدالت است سعی بر این داشته‌ایم تا روش‌های موجود را به‌دقت مورد بررسی و مطالعه قرار دهیم و به تحلیل نقاط قوّت، محدودیت‌ها و کاربردهای آن‌ها بپردازیم. در نهایت بر مبنای نتایج مطالعات و تحلیل‌های صورت گرفته، برخی از چالش‌ها و مسائل باز مرتبط را معرفی کرده‌ایم.

در راستای اهداف پیش گفته‌شده، در این مقاله در قالب یک چارچوب سلسله‌مراتبی ابتدا روش‌های اندازه‌گیری قابلیت استفاده مجدد به دو گروه عمده‌ی سنجه‌ها[۶] و مدل‌ها[۷] تقسیم‌بندی می‌شوند. سپس در سطح بعد، هرکدام از این گروه‌ها خود به سه زیرگروه به نام روش‌های مبتنی بر جعبه سیاه، روش‌های مبتنی بر جعبه سفید روش‌های مبتنی بر جعبه خاکستری تقسیم می‌گردند این سلسله‌مراتب تا رسیدن به برگ‌ها ادامه پیدا می‌کند.

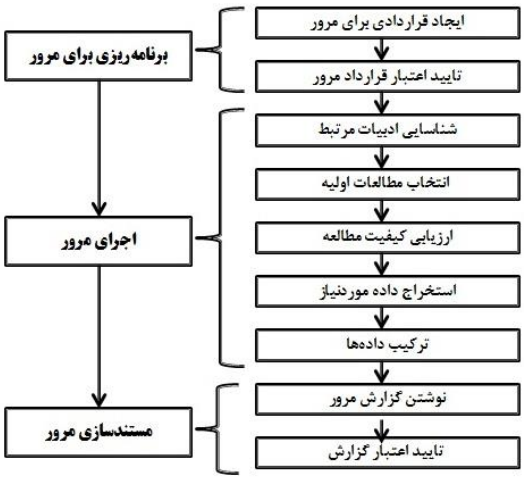
ادامه‌ی مقاله به این ترتیب سازمان‌دهی شده‌است: در بخش دوم به بیان روش انجام تحقیق خواهیم‌پرداخت. در بخش سوم سنجه‌ها و مدل‌های ارائه شده در رابطه با قابلیت استفاده‌ی مجدد مؤلفه را معرفی خواهیم‌کرد. در بخش چهارم مرور صورت گرفته در این مطالعه را با دیگر مرورهای موجود مقایسه خواهیم‌نمود. در بخش پنجم به بررسی چالش‌ها و مسائل باز موجود خواهیم‌پرداخت و در نهایت در بخش ششم نتیجه‌گیری از موضوعات مطروحه پیشنهاد کارهایی برای آینده را خواهیم‌داشت.

۲- روش انجام تحقیق

در این مقاله برای دریافت درک درستی از تعاریف و روش‌های ارائه‌شده برای ارزیابی[۸] و اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد، مروری نظام‌مند بر ادبیات گذشته انجام داده‌ایم. یک مرور نظام‌مند برای تولید خروجی معتبر، منصفانه، عادلانه و قابل‌اعتماد از روش‌های شفاف و قابل تکرار کمک می‌گیرد. در این نوع مرور، ساختار حاکم بر فرآیند تحقیق، مانع از اعمال نظرات شخصی، پیش‌داوری و مهندسی نتایج به سمت نتایج دلخواه پژوهشگر می‌گردد [۴۳-۴۸]. در شکل (۱) ساختار کلی یک مرور نظام‌مند نشان داده شده‌است.

در این مقاله با کلماتی کلیدی مانند: ارزیابی/ اندازه‌گیری کمیّ قابلیت استفاده مجدد نرم‌افزار، سنجه‌ها/ مدل‌های قابلیت استفاده مجدد برای سیستم‌های مبتنی بر مؤلفه و در منابعی نظیر: «ACM Digital Library»، «Scopus»، « Springer Online Journal Collection»، «Pad - پایگاه اشتراک داده دانشگاه فردوسی مشهد»، «Science Direct»، « Xplore IEEE Digital Library» و نیز «Google Scholar» به مرور نظام‌مند تحقیقات صورت پذیرفته در حوزه مربوطه پرداخته‌ایم و در فرآیند تحقیق، منابع (نظیر

مقالات، کتب، پایان‌نامه‌ها و رساله‌ها) به زبان انگلیسی و فارسی را مورد توجه قرار داده‌ایم. البته بخش عمده‌ای از منابع مورد تحقیق را مقالات معتبر تشکیل داده‌اند. برای تعیین این‌که آیا مقالات بدست آمده حاوی اطلاعات مرتبط هستند، عنوان و چکیده‌ی آن‌ها را مورد مطالعه قرار داده‌ایم که در نهایت ۲۷ مقاله برای مطالعه‌ی بیشتر انتخاب شده‌است.



شکل (۱): ساختار کلی یک مرور نظام‌مند [۴۳]

۳- معرفی روش‌های مختلف اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد مؤلفه‌های نرم‌افزاری

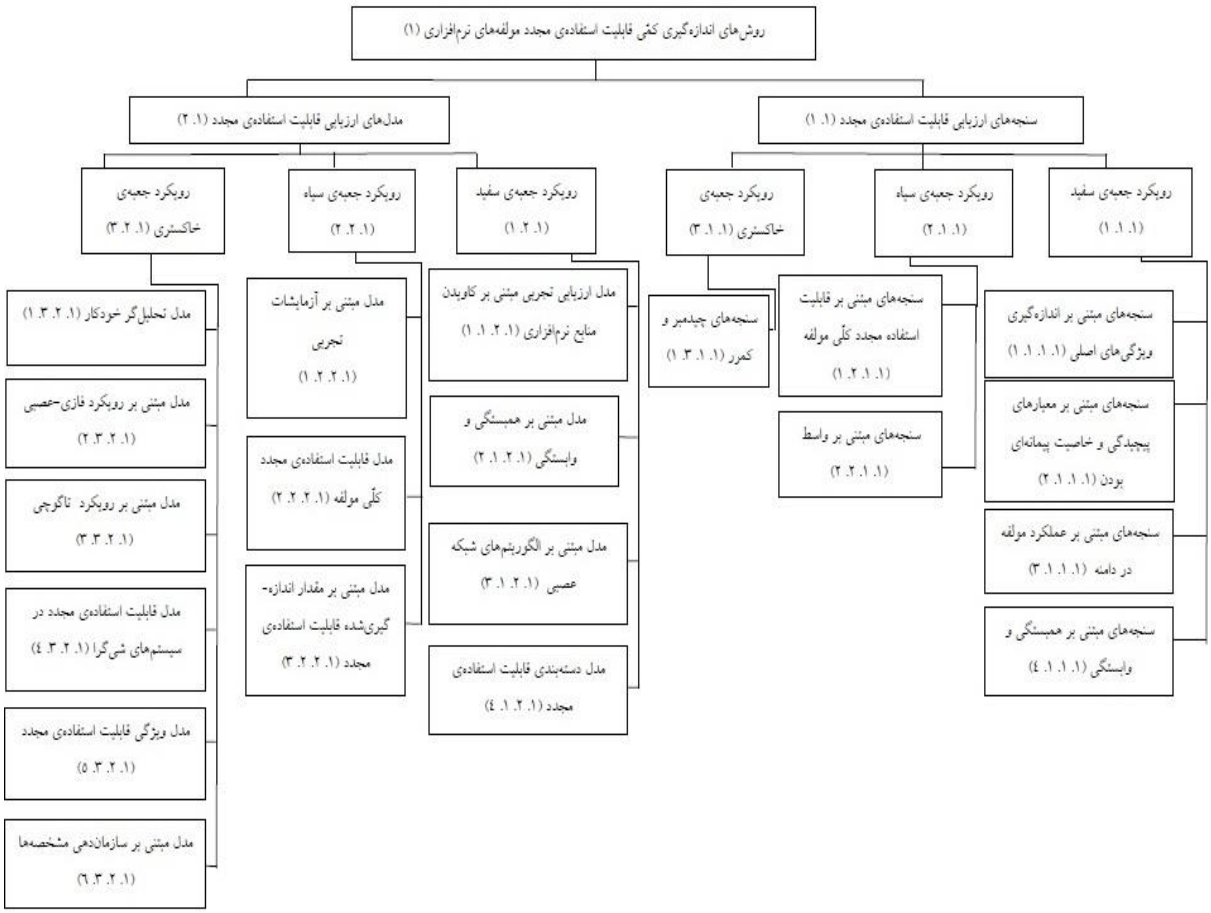
تاکنون روش‌های مختلفی برای اندازه‌گیری کمّی قابلیت استفاده مجدد مؤلفه‌های نرم‌افزاری ارائه شده‌است. در این مقاله با ارائه‌ی یک چارچوب جدید، که آن‌را درخت روش-های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد نامیده‌ایم، این روش‌ها به‌صورت کاملی دسته‌بندی گردیده‌اند. در شکل (۲) درخت روش‌های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد نشان داده شده‌است. همانطور که در شکل می‌بینیم در سطح اول از این چارچوب سلسله-مراتبی، روش‌های موجود به دو گروه کلّی سنجه‌ها و مدل‌ها تقسیم‌بندی شده‌اند. در سطح بعد، هرکدام از این گروه‌ها خود به سه زیرگروه به نام روش‌های مبتنی بر جعبه سیاه، روش‌های مبتنی بر جعبه سفید و روش‌های مبتنی بر جعبه خاکستری تقسیم می‌شوند و به همین ترتیب این سلسله‌مراتب تا رسیدن به برگ‌ها ادامه پیدا می‌کند. لازم به ذکر است که برای سهولت پیگیری روش-ها در بخش‌های بعدی، به هر کدام از گره‌های درخت شماره‌ای تعلّق گرفته‌است و در ادامه، ارجاع به هر یک از گره‌های درخت از طریق شماره‌ی آن صورت می‌گیرد.

۱-۳- دسته ۱. ۱: سنجه‌های ارزیابی قابلیت استفاده‌ی مجدد

سنجه، شاخصی کمّی از یک فاکتور کیفی است و یکی از مهمترین ابزارهای ارزیابی کمّی قابلیت استفاده‌ی مجدد از نرم‌افزار محسوب می‌شود. در این بخش مروری بر سنجه-های ارائه شده در رابطه با قابلیت استفاده‌ی مجدد انجام داده‌ایم و همانطور که گفته شد آن‌ها را به سه دسته‌ی کلی تقسیم کرده‌ایم: ۱- سنجه‌های جعبه سفید، ۲- سنجه‌های جعبه سیاه و ۳- سنجه‌های جعبه خاکستری. در هر سه دسته‌ی ارائه شده در این بخش، با هدف نشان دادن میزان پیشرفت در اندازه‌گیری قابلیت استفاده‌ی مجدد در طی زمان، نتایج تحقیقات را به ترتیب زمانی پشت سرهم آورده‌ایم.

۱-۱-۳ زیر دسته ۱. ۱. ۱: رویکرد جعبه‌ی سفید

در رویکرد جعبه‌ی سفید، توسعه‌دهنده به کد مؤلفه دسترسی دارد و برای تطبیق بیشتر مؤلفه با نیازمندی‌های محیط جدید، می‌تواند برروی مؤلفه تغییراتی را اعمال کند. به همین دلیل جعبه‌ی سفید رویکردی انعطاف‌پذیر می‌باشد. اما یکی از مهم‌ترین معایب این روش این است که در آن برای تطبیق بیشتر مؤلفه با محیط جدید کاربر باید از جزئیات پیاده‌سازی مؤلفه آگاه باشد. در ادامه و در این بخش، تعدادی از مهمترین سنجه‌های ارائه شده در رابطه با اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد از مؤلفه با این رویکرد را عنوان خواهیم‌کرد.



شکل (۲): درخت روش‌های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد مؤلفه‌ی نرم‌افزار

زیر دسته ۱. ۱. ۱. ۱: سنجه‌های مبتنی بر اندازه‌گیری ویژگی‌های اصلی

در سال ۱۹۹۱، کالدریا و باسیلی [۱۴] سه ویژگی اصلی ولی انتزاعی را برای ارزیابی قابلیت استفاده‌ی مجدد معرفی کردند: ۱-هزینه‌های استفاده مجدد[۹]، ۲- کارایی عملکردی[۱۰] ۳- کیفیت مؤلفه‌ها[۱۱]. این ویژگی‌ها به‌وسیله‌ی فاکتورهایی تعیین می‌گردند که به طور مستقیم و یا غیرمستقیم توسط سنجه‌های کلاسیک مانند حجم هالستید[۱۲] و پیچیدگی سایکلوماتیک مک-کِیب[۱۳] و یا سنجه‌هایی نظیر قائده‌مندی[۱۴] و بسامد استفاده‌ی مجدد[۱۵] اندازه‌گیری می‌شوند.

سنجه حجم به‌منظور تخمین دو ویژگی اول، سنجه پیچیدگی سایکلوماتیک برای ارزیابی هر سه ویژگی، سنجه قائده‌مندی برای ارزیابی ویژگی‌های دوم و سوم و در نهایت سنجه بسامدِ استفاده‌ی مجدد صرفا برای ارزیابی کارایی عملکردی، مورد استفاده قرار می‌گیرند. در ادامه مروری مختصر بر روی هریک از این سنجه‌ها خواهیم‌داشت.

زیر دسته ۱. ۱. ۱. ۱. ۱: سنجه حجم هالستید

معیارهای پیچیدگی هالستید در سال ۱۹۷۷ توسط هُوارد هالستید [۱۵] و به عنوان بخشی از رساله‌ی دکترای وی معرفی گردیدند. او ادعا کرد که سنجه‌های نرم‌افزاری باید پیاده‌سازی و یا بیان الگوریتم‌ها در زبان‌های مختلف را منعکس کنند اما مستقل از اجرا برروی یک پلت فرم خاص باشند. در واقع هدف او شناسایی ویژگی‌های قابل اندازه‌گیری از نرم‌افزار و بیان روابط بین آن‌ها بود. این کار شبیه به شناسایی خواص قابل اندازه‌گیری از ماده (مانند حجم، جرم و فشار یک گاز خاص) و روابط بین آن‌ها (مشابه معادله‌ی گاز) است.

هالستید عنوان کرد که اگر برای یک مسئله‌ی داده شده،

n_1 تعداد عملگرهای مجزا[16]، n_2 تعداد عملوندهای مجزا[17]، N_1 تعداد کل عملگرها و N_2 تعداد کل عملوندها باشد، در این‌صورت به وسیله‌ی این اعداد معیارهای مختلفی نظیر معیارهای زیر می‌توانند محاسبه شوند.

- واژگان برنامه[18]:

$$n = n_1 + n_2 \quad (1)$$

- طول برنامه[19]:

$$N = N_1 + N_2 \quad (2)$$

- تخمینی از طول برنامه[20]:

$$\hat{N} = n_1 log_2 n_1 + n_2 log_2 n_2 \quad (3)$$

- حجم[21] هالستید:

$$V = N \times \log_2 n \quad (4)$$

- دشواری فهم و یا نوشتن یک برنامه[22]:

$$D = \frac{n_1}{2} \times \frac{N_2}{n_2} \quad (5)$$

زیر دسته ۱. ۱. ۱. ۱. ۲: سنجه قائده‌مندی

سنجه‌ی قائده‌مندی براساس مفهوم پیش‌بینی طول برنامه برحسب یکسری فرضیات، استوار است [۱۴]. همانطور که در بخش قبل اشاره کردیم طول واقعی یک برنامه را می‌توان از طریق رابطه (۲) و طول تخمینی از برنامه را می‌توان از طریق رابطه (۳) محاسبه‌کرد. در این‌صورت سنجه قائده‌مندی میزان درستی تخمین صورت گرفته را نشان خواهدداد و از طریق رابطه (۶) قابل محاسبه است.

$$\text{Regularity} = 1 - \{\frac{N-\hat{N}}{N}\} \quad (6)$$

زیر دسته ۱. ۱. ۱. ۱. ۳: سنجه بسامدِ استفاده‌ی مجدد

اگر تعداد فراخوانی‌های ایستا به یک مؤلفه را با تعداد فراخوانی‌های کلاسی از مؤلفه‌ها که قابل استفاده‌ی مجدد می‌باشند مورد مقایسه قرار دهیم، در این‌صورت می‌توانیم بسامد استفاده‌ی مجدد از آن مؤلفه را تخمین بزنیم [۱۴]. فرض کنید سیستم ما ترکیبی باشد از مؤلفه‌های $C_1, \ldots, C_N$ که توسط کاربر تعریف شده‌اند و مؤلفه‌های $S_1, \ldots, S_M$ که در محیطی استاندارد تعریف شده‌اند (مثل printf در زبان C). اگر $n(C)$ تعداد فراخوانی‌های مؤلفه‌ی داده شده‌ی C باشد، در این‌صورت استفاده‌ی مجدد این مؤلفه به‌صورت ایستا و طبق رابطه‌ی (۷) قابل محاسبه است.

$$v_\sigma(C) = \frac{n(C)}{\frac{1}{M}\sum_{i=1}^{M} n(S_i)} \quad (7)$$

که در آن مخرج کسر عبارت است از تعداد متوسط فراخوانی‌های یک مؤلفه‌ی استاندارد.

زیر دسته ۱. ۱. ۱. ۱. ۴: سنجه پیچیدگی سایکلوماتیک مک-کِیب

در سال ۱۹۷۶ مک-کِیب سنجه‌ای را برای اندازه‌گیری پیچیدگی در نرم‌افزار به نام پیچیدگی سایکلوماتیک معرفی کرد [۱۶]. این سنجه در کد منبع یک برنامه، تعداد مسیرهایی را که به‌صورت خطی از یکدیگر مستقل هستند اندازه‌گیری می‌کند. پیچیدگی سایکلوماتیک با استفاده از گراف کنترل جریان یک برنامه محاسبه می‌گردد. گراف کنترل جریان، گرافی جهت‌دار است که شامل بلوک‌های اصلی یک برنامه می‌شود و دو بلوک با یک یال به‌هم متصل هستند اگر امکان انتقال کنترل از بلوک اول به بلوک دوم وجود داشته باشد. در این صورت سنجه پیچیدگی را از طریق رابطه (۸) می‌توان محاسبه کرد.

$$M = E - N + 2 \quad (8)$$

که در آن E تعداد یال‌های گراف، N تعداد گره‌های گراف و P تعداد مؤلفه‌های متصل در گراف است.

زیر دسته ۱. ۱. ۱. ۲: سنجه‌های مبتنی بر معیارهای پیچیدگی و خاصیت پیمانه‌ای بودن

در سال ۲۰۰۰، لی و چنگ مجموعه‌ی دیگری از سنجه‌ها را برای اندازه‌گیری قابلیت استفاده‌ی مجدد و قابلیت نگهداری از نرم‌افزارهای شی‌گرا ارائه کردند [۱۸]. معیارهای استفاده‌شده، پیچیدگی و خاصیت پیمانه‌ای بودن هستند که سنجه‌های متناظر با آن‌ها به‌ترتیب پیچیدگی داخلی[23]، پیچیدگی خارجی[24]، وابستگی[25]

همبستگی[۲۶] در کلاس‌ها می‌باشند. در ادامه مروری اجمالی بر روی هر یک از این سنجه‌ها خواهیم‌داشت.

زیر دسته ۱. ۱. ۱. ۲. ۱: سنجه پیچیدگی داخلی کلاس

سنجه پیچیدگی داخلی کلاس (ICC) ترکیبی از پیچیدگی صفت‌های کلاس (CAC) و پیچیدگی متدهای کلاس (CMC) می‌باشد و از طریق رابطه (۹) قابل محاسبه است.

$$ICC = w_{cac}CAC + w_{cmc}CMC \qquad (۹)$$

که در آن w_{cac} و w_{cmc}، وزنِ در نظر گرفته شده هستند. با درنظرگرفتن اینکه صفت می‌تواند یک کلاس و یا یک اعلان نوع باشد، پیچیدگی صفت‌های کلاس را می‌توان به‌وسیله‌ی رابطه (۱۰) بدست آورد.

$$CAC = \sum_i^{NA} AC_i \qquad (۱۰)$$

که در آن NA تعداد صفت‌ها در یک کلاس است و پیچیدگی یک صفت (AC) را با توجه به جدول (۲) می‌توان محاسبه کرد.

پیچیدگی متدهای کلاس را می‌توان به‌وسیله‌ی رابطه (۱۱) بدست آورد.

$$CMC = \sum_i^{NM} MC_i \qquad (۱۱)$$

که در آن NM تعداد متدها در یک کلاس است پیچیدگی یک متد (MC) همان پیچیدگی ظاهری[۲۷] متد در نظر گرفته شده‌است. سنجه پیچیدگی ظاهری، پیچیدگی واحدهای قابل اندازه‌گیری در متد را محاسبه می‌کند و یک متد می‌تواند چندین واحد قابل اندازه‌گیری داشته باشد.

زیر دسته ۱. ۱. ۱. ۲. ۲: سنجه پیچیدگی خارجی کلاس

سنجه پیچیدگی خارجی کلاس (ECC) صفت‌ها و متدهای عمومی و همچنین پیچیدگی پارامترهای مربوط به متدهای عمومی از یک کلاس را مد نظر قرار می‌دهد. این سنجه از طریق رابطه (۱۲) قابل محاسبه است.

$$ECC = w_{cic}CIC + w_{ipc}IPC \qquad (۱۲)$$

که در آن CIC و IPC به‌ترتیب بیانگر پیچیدگی واسط کلاس و پیچیدگی پارامتر واسط می‌باشند و w_{cic} و w_{ipc}، وزنِ در نظر گرفته شده هستند.

پیچیدگی واسط کلاس، برای اندازه‌گیری معیار دسترسی مورد استفاده قرار می‌گیرد و از طریق رابطه (۱۳) بدست می‌آید.

$$CIC = \sum_i^{NPA} AC_i + \sum_i^{NPM} MC_i \qquad (۱۳)$$

که در آن NPA و NPM به‌ترتیب بیانگر تعداد صفت‌ها و متدهای عمومی هستند و منظور از AC و MC دقیقا همان چیزی است که قبلا در بخش پیچیدگی داخلی کلاس به آن اشاره کرده‌ایم.

پیچیدگی پارامتر واسط، از طریق رابطه (۱۴) قابل محاسبه است.

$$IPC = \sum_i^{NPM} PC_i \qquad (۱۴)$$

که در آن PC همان پیچیدگی از پیش تعیین شده برای صفت‌ها می‌باشد و با توجه به نوع پارامترهای مربوط به متدهای عمومی و از طریق جدول (۲) قابل محاسبه است.

جدول (۲): پیچیدگی از پیش تعیین شده برای صفت‌ها [۱۸]

نوع داده	پیچیدگی
Int	۱
Short	۲
Long	۲
Byte	۲
Char	۲
String	۳
Boolean	۱
Float	۳
Double	۳
Class	۶

زیر دسته ۱. ۱. ۱. ۲. ۳: سنجه پیچیدگی همبستگی در یک کلاس

این سنجه معیاری برای اندازه‌گیری میزان ارتباط بین مؤلفه‌ها، صفت‌ها و متدهای یک کلاس را ارائه می‌کند و از طریق رابطه (۱۵) قابل محاسبه است.

$$CHC = w_{invm}INVM/ALLMC + .w_{shm}SHM/ALLMC \quad (۱۵)$$

که در آن INVM تعداد متدهایی است که متدهای تعریف شده در کلاس را صدا می‌زنند، SHM تعداد متدهایی است که صفت‌هایی را با متدهای دیگری که در کلاس‌های دیگر تعریف شده‌اند به اشتراک می‌گذارند و ALLMC تعداد کل متدهایی است که در کلاس تعریف شده‌اند.

زیر دسته ۱. ۱. ۱. ۲. ۴: سنجه پیچیدگی وابستگی در یک کلاس

این سنجه معیاری را برای اندازه‌گیری میزان وابستگی یک کلاس به دیگر کلاس‌های موجود در سیستم، ارائه می‌کند و از طریق رابطه (۱۶) قابل محاسبه است. مقادیر بالای این سنجه برای یک کلاس بیانگر نیاز به تلاش بیشتر برای فهم و تغییر آن کلاس است.

$$CPC = w_{outm}OUTM/ALLM + . \quad w_{inm}INM/ALLM \quad (۱۶)$$

که در آن OUTM تعداد متدهایی است که متدهای تعریف شده در خارج از کلاس فراخوانی شده را صدا می‌زنند، INM تعداد متدهایی است که خارج از کلاس تعریف شده‌اند و متدهایی را که در داخل کلاس تعریف شده‌اند صدا می‌زنند و ALLM تعداد کل متدهای فراخوانی‌شده‌ای است که در سیستم تعریف شده‌اند.

زیر دسته ۱. ۱. ۱. ۳: سنجه‌های مبتنی بر عملکرد مؤلفه در دامنه

در سال ۲۰۰۱، چو و همکارانش برای پیچیدگی مؤلفه، قابلیت سفارشی‌سازی، استفاده‌ی مجدد و قابلیت استفاده‌ی مجدد سنجه‌هایی را ارائه کردند [۱۹]. در مقاله آنها، قابلیت استفاده‌ی مجدد یک مؤلفه به‌وسیله‌ی عملکردی که مؤلفه‌های نرم‌افزاری در دامنه‌شان دارند مشخص می‌گردد و به‌صورت نسبت بین تعداد متدهای واسط در مؤلفه که در دامنه‌ی خود توابع اشتراک[۲۸] را ارائه می‌کنند به تعداد کل متدهای واسط در مؤلفه تعریف می‌شود. هرچقدر یک مؤلفه توابع اشتراک بیشتری را فراهم کند، قابلیت استفاده‌ی مجدد از آن بالاتر می‌رود. علاوه‌بر سنجه یادشده، آن‌ها سنجه‌هایی را نیز برای قابلیت سفارشی‌سازی مؤلفه پیشنهاد کردند و مدعی شدند که اگر امکان سفارشی‌سازی به یک مؤلفه داده نشود در این‌صورت قابلیت استفاده‌ی مجدد از آن پایین می‌آید. در مقاله‌ی ارائه شده توسط آن‌ها قابلیت سفارشی‌سازی از طریق سنجه تنوع مؤلفه[۲۹] تعیین می‌شود که این سنجه به‌صورت نسبت بین تعداد متدهای سفارشی‌سازی شده به کل متدهای موجود در واسط‌های یک مؤلفه تعریف می‌گردد. در ادامه هریک از این سنجه‌ها را به اختصار مرور خواهیم‌کرد.

زیر دسته ۱. ۱. ۱. ۳. ۱: سنجه تنوع مؤلفه

این سنجه برای اندازه‌گیری قابلیت سفارشی‌سازی که به صورت غیرمستقیم بر روی قابلیت استفاده‌ی مجدد تاثیر می‌گذارد ارائه شده‌است و از طریق رابطه (۱۷) قابل محاسبه است:

$$CV = \frac{\sum_{i=1}^{m} Count(CVM_i)}{\sum_{j=1}^{m} Count(CIM_j)} \quad (۱۷)$$

که در آن Count(CVM) تعداد متدهای سفارشی‌سازی شده و Count(CIM) تعداد متدهای اعلان شده در هر واسط می‌باشد.

مقدار CVM را می‌توان از طریق رابطه (۱۸) محاسبه کرد:

$$CVM = \sum_{i=1}^{m} Count(CVMa_i) + \sum_{j=1}^{n} Count(CVMm_j) + \sum_{k=1}^{u} Count(CVMw_k) \quad (۱۸)$$

که در آن Count(CVMa) تعداد متدها در سفارشی‌سازی صفت[۳۰]، Count(CVMm) تعداد متدها در سفارشی‌سازی رفتار[۳۱] و Count(CVMw) تعداد متدها در سفارشی‌سازی گردش کاری[۳۲] است.

زیر دسته ۱. ۱. ۱. ۳. ۲: سنجه قابلیت استفاده‌ی مجدد از مؤلفه

سنجه قابلیت استفاده‌ی مجدد از مؤلفه را از طریق رابطه (۱۹) می‌توان محاسبه کرد:

$$CR = \frac{\sum_{i=1}^{n} Count(CCM_i)}{\sum_{j=1}^{m} Count(CIM_j)} \quad (۱۹)$$

که در آن Count(CCM) تعداد متدهای واسط که عملکردهایی مشترک را میان چندین کاربرد در یک دامنه فراهم می‌آورند و Count(CIM) تعداد کل متدهای اعلان شده در واسط‌های یک مؤلفه است.

زیر دسته ۱. ۱. ۱. ۴: سنجه‌های مبتنی بر پیوستگی[۳۳] و وابستگی[۳۴]

در سال ۲۰۰۸، گوی و اسکات به‌منظور اندازه‌گیری قابلیت استفاده‌ی مجدد مؤلفه‌های جاوا، فرمول‌های اصلاح‌شده‌ای را برای محاسبه‌ی سنجه‌های پیوستگی و وابستگی ارائه کردند [۲۰]. به غیر از سنجه‌های CBO، RFC و LCOM که قبلا آن‌ها را توضیح داده‌ایم تعدادی دیگر از سنجه‌های پایه در محاسبه‌ی وابستگی و پیوستگی وجود دارند که در جدول (۳) آن‌ها را به اختصار بیان کرده‌ایم. فرمول‌های اصلاح‌شده‌ی وابستگی و پیوستگی ارائه شدند تا بگویند کلاس‌ها تا چه حد بهم وابسته هستند و متدها تا چه حد باهم همبستگی دارند. علاوه-براین، آن‌ها با مورد توجه قرار دادن روابط تعدی[۳۵]، دو سنجه دیگر را برای اندازه‌گیری قابلیت استفاده‌ی مجدد از نرم‌افزار پیشنهاد کردند که این سنجه‌ها بر پایه‌ی سنجه-های همبستگی و وابستگی که خودشان ارائه کرده بودند بنا شده‌است. نویسندگان این مقاله اذعان کرده‌اند، در تعیین قابلیت استفاده‌ی مجدد یک مؤلفه موارد دیگری نیز نقش دارند که در این مقاله مورد توجه قرار نگرفته‌اند.

زیر دسته ۱. ۱. ۱. ۴. ۱: سنجه وابستگی

هر سیستم نرم‌افزاری شی‌گرا را می‌توان به‌صورت یک گراف جهت‌دار که در آن رئوس، کلاس‌های تشکیل‌دهنده‌ی سیستم هستند، درنظر گرفت. فرض کنید چنین سیستمی شامل مجموعه کلاس‌های $C = \{C_1, C_2, \dots, C_m\}$ باشد، در این‌صورت اگر $M_j = \{M_{j,1}, M_{j,2}, \dots, M_{j,n}\}$ نمایانگر متدهای کلاس C_j و برای $j \neq i$ ، $R_{j,i}$ نمایانگر مجموعه متدها و متغیرهای نمونه در کلاس C_i که توسط کلاس C_j فراخوانی می‌شوند، باشد در این‌صورت از C_j به C_i یال وجود دارد اگر و فقط اگر $R_{j,i}$، تهی نباشد. بنابراین یال‌های گراف بیانگر وابستگی مستقیم یک کلاس به کلاس دیگر هستند. گراف مذکور جهت‌دار است چراکه لزوما $R_{j,i}$ با $R_{i,j}$ برابر نیست. با توجه به توضیحات داده‌شده، مجموعه‌ی همه‌ی متدها و متغیرهای نمونه در دیگر کلاس‌ها که توسط کلاس C_j فراخوانی می‌شوند را می‌توان از طریق رابطه (۲۰) به‌دست آورد:

$$R_j = U_{1 \le i \le m} R_{j,i} \quad (۲۰)$$

در گام بعد باید به هر یال، عددی را نسبت بدهیم که بیانگر میزان وابستگی مستقیم یک کلاس به کلاس دیگر باشد. به وضوح هر چقدر یک کلاس متدهای بیشتری از کلاس‌های دیگر را صدا بزند این عدد نیز به همان نسبت بزرگتر خواهد بود. البته این عدد باید بازگو کننده‌ی این حقیقت نیز باشد که اگر کلاسی متدهای زیادی را فراخوانی کند در این‌صورت نسبت به دیگر کلاس‌ها، متدهای فراخوانی کننده‌ی بالاتری دارد. با توجه به توضیحات گفته شده، سنجه $CoupD(i,j)$ که گویای وابستگی مستقیم کلاس C_i به C_j است به‌صورت رابطه (۲۱) تعریف شده‌است.

$$CoupD(i,j) = \frac{|R_{i,j}|}{|R_i| + |M_i|} \quad (۲۱)$$

که در آن، مخرج کسر بیانگر تعداد کل متدهای فراخوانی شده توسط کلاس C_i است و بنابراین:

$$0 \le CoupD(i,j) \le 1$$

حال به وابستگی غیر مستقیم بین کلاس‌ها می‌پردازیم. فرض کنید $CoupD(i,j)$ و $CoupD(j,k)$ دارای مقادیر متناهی باشند ولی مقدار $CoupD(i,k)$ برابر باصفر باشد. بنابراین با وجود اینکه هیچ وابستگی مستقیمی بین کلاس‌های C_i و C_k وجود ندارد ولی وابستگی از نوع غیرمستقیم را داریم چراکه C_i متدهای C_j را فراخوانی می‌کند که خود کلاس C_j متدهای C_k را صدا می‌زند.

جدول (۳): سنجه‌های پایه در محاسبه‌ی وابستگی و پیوستگی [۲۰]

نام سنجه	نوع سنجه	تعریف
LCOM3	همبستگی	تعداد مؤلفه‌های متصل در گرافی که رئوسش متدها هستند و بین دو متد یال وجود دارد اگر و فقط اگر مشابه یکدیگر باشند.
RLCOM	همبستگی	نسبت بین تعداد جفت متدهای غیر مشابه به کل تعداد جفت متدها در کلاس
TCC	همبستگی	نسبت بین تعداد جفت متدهای مشابه به کل تعداد جفت متدها در کلاس
CF	وابستگی	دو کلاس را وابسته به هم گوییم اگر متدها یا متغیرهای نمونه در یک کلاس توسط دیگری استفاده شوند. در این‌صورت CF در یک نرم‌افزار تعداد جفت کلاس‌های وابسته به هم تقسیم بر تعداد کل جفت کلاس‌ها خواهد بود.
DAC	وابستگی	تعداد صفت‌هایی که دیگر کلاس‌ها را به عنوان نوع‌های خود استفاده می‌کنند.

از آن‌جا که میزان این وابستگی به دو وابستگی مستقیم ذکر شده بستگی دارد بنابراین یک معیار معقولانه، ضرب این دو در یکدیگر است. این مفهوم را می‌توان به‌سادگی تعمیم داد. دو کلاس با یکدیگر وابستگی دارند اگر از یکی از آن‌ها مسیری به دیگری وجود داشته باشد و مقادیر $CoupD$ یال‌های شرکت کننده در آن مسیر همگی غیرصفر باشند، در این صورت مقدار وابستگی را می‌توان از ضرب مقادیر $CoupD$ها در یکدیگر به‌دست آورد. باتوجه به توضیحات داده شده، سنجه $CoupT(i,j,\pi)$ که بیانگر وابستگی مبتنی بر رابطه‌ی تعدی بین کلاس‌های C_i و C_j، در مسیر مشخص π می‌باشد را می‌توان از طریق رابطه (۲۲) محاسبه کرد.

$$CoupT(i,j,\pi) = \prod_{e_{s,t}\in\pi} CoupD(s,t) = \prod_{e_{s,t}\in\pi} \frac{|R_{s,t}|}{|R_s|+|M_s|} \quad (۲۲)$$

که در آن $e_{s,t}$ بیانگر یال موجود بین رئوس s و t است. به‌طور کلی ممکن است بین دو کلاس بیشتر از یک مسیر وجود داشته باشد که همگی آن‌ها مقدار $CoupT$ مخالف صفری داشته باشند، در این‌صورت طبق رابطه (۲۳) یک راهکار عملی این است که مسیری با بیشترین مقدار $CoupT$ را انتخاب کنیم.

$$Coup(i,j) = CoupT(i,j,\pi_{max}) \quad (۲۳)$$

که در آن

$$\pi_{max}(i,j) = argmax_{\pi\in\Pi} CPT(i,j,\pi) \quad (۲۴)$$

که در آن Π مجموعه‌ی تمام مسیرها از C_i به C_j می‌باشد. حال با داشتن معیاری برای میزان وابستگی بین جفت کلاس‌ها، طبق رابطه (۲۵) از این معیار می‌توان به‌عنوان پایه‌ای برای اندازه‌گیری وابستگی کلی یک سیستم نرم‌افزاری استفاده کرد.

$$WTCoup = \frac{\sum_{i,j=1}^{m} Coup(i,j)}{m^2-m} \quad (۲۵)$$

که در آن m تعداد کلاس‌های موجود در سیستم است.

زیر دسته ۱. ۱. ۱. ۴. ۲: سنجه همبستگی

همبستگی کلاس‌ها بر مبنای میزان شباهت متدها تعریف می‌شود. دو متد مشابه یکدیگر هستند اگر مجموعه متغیرهای نمونه‌ای که به آن‌ها دسترسی دارند با یکدیگر اشتراک داشته باشد. در اینجا نیز مانند معیار وابستگی از دیدگاه نظریه‌ی گراف استفاده شده‌است. در این مورد متدهای کلاس، رئوس گراف ما هستند.

فرض کنید یک کلاس دارای مجموعه‌ای از متدها به‌صورت $M = \{M_1, M_2, \dots, M_m\}$ باشد و $V_j = \{V_{j,1}, V_{j,2}, \dots, V_{j,n}\}$ متغیرهای نمونه‌ای باشند که توسط متد M_j قابل دسترس هستند. در این‌صورت از M_i به M_j یال وجود دارد اگر و فقط اگر $V_j \cap V_i$ تهی نباشد. برخلاف گراف وابستگی، گراف شباهت غیرجهت‌دار است چراکه اشتراک یک رابطه‌ی تقارنی است.

در گام بعد، به هر یال عددی را نسبت می‌دهیم که بیانگر میزان متغیرهای نمونه‌ای مشترکی بین دو متد می‌باشد. با توجه به توضیحات گفته شده، سنجه $SimD(i,j)$ که شباهت مستقیم بین دو متد M_i و M_j را اندازه‌گیری می‌کند، به‌صورت رابطه (۲۶) تعریف شده‌است.

$$SimD(i,j) = \frac{|V_j \cap V_i|}{|V_j \cup V_i|} \quad (۲۶)$$

که در آن $i \neq j$ و $0 \leq SimD(i,j) \leq 1$

حال به شباهت غیر مستقیم بین دو متد می‌پردازیم. در گراف شباهت میزان شباهت بین دو متد، برابر است با حاصل‌ضرب مقادیر $SimD$ از یال‌های تشکیل‌دهنده‌ی مسیر موجود بین آن دو متد. بنابراین برای یک مسیر مشخص مانند π شباهت مبتنی بر رابطه‌ی تعدی بین متدهای M_i و M_j را می‌توان بر طبق رابطه (۲۷) به‌دست آورد.

$$SimT(i,j,\pi) = \prod_{e_{s,t}\in\pi} SimD(s,t) = \prod_{e_{s,t}\in\pi} \frac{|V_s \cap V_t|}{|V_s \cup V_t|} \quad (۲۷)$$

که در آن، $e_{s,t}$ بیانگر یال موجود بین رئوس s و t است.

در اینجا نیز برطبق رابطه (۲۸) در صورت وجود چند مسیر، مسیری با بیشترین مقدار $SimT$ بیانگر شباهت موجود بین دو متد خواهد بود.

$$Sim(i,j) = SimT(i,j,\pi_{max}) \quad (۲۸)$$

که در آن

$$\pi_{max}(i,j) = arg\ max_{\pi\in\Pi} SimT(i,j,\pi) \quad (۲۹)$$

حال با داشتن معیاری برای میزان شباهت بین جفت متدها، طبق رابطه (۳۰) از این معیار می‌توان به‌عنوان پایه‌ای برای اندازه‌گیری همبستگی یک کلاس استفاده کرد.

$$ClassCoh = \frac{\sum_{i,j=1}^{m} Sim(i,j)}{m^2 - m} \quad (۳۰)$$

که در آن m تعداد متدهای موجود در کلاس است.

در نهایت برای یک سیستم نرم‌افزاری کامل، همبستگی وزن‌دار مبتنی بر رابطه‌ی تعدی به‌صورت رابطه (۳۱) تعریف شده‌است.

$$WTCoh = \frac{\sum_{j=1}^{n} ClassCoh_j}{n} \quad (۳۱)$$

که در آن n تعداد کلاس‌های موجود در سیستم است.

۲-۱-۳ زیر دسته ۱. ۱. ۲: رویکرد جعبه‌ی سیاه

در استفاده‌ی مجدد مؤلفه با رویکرد جعبه‌ی سیاه ساختار داخلی مؤلفه برای کاربر قابل دسترس نمی‌باشد و امکان تغییر کد مؤلفه وجود ندارد. در رویکرد جعبه‌ی سیاه اطلاعات مورد نیاز برای اندازه‌گیری از بخش خارجی مؤلفه بدست می‌آید. در ادامه و در این بخش، تعدادی از مهمترین سنجه‌های ارائه شده در رابطه با اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد از مؤلفه با رویکرد مذکور را عنوان خواهیم‌کرد.

زیر دسته ۱. ۱. ۲. ۱: سنجه‌های مبتنی بر قابلیت استفاده مجدد کلی مؤلفه

اولین مجموعه‌ی سنجه‌های اندازه‌گیری قابلیت استفاده‌ی مجدد از مؤلفه‌های جعبه‌ی سیاه در سال ۲۰۰۳ توسط واشیزاکی و همکارانش پیشنهاد شد [۵]. این سنجه‌ها سه فاکتور مهم و تاثیرگذار بر روی قابلیت استفاده‌ی مجدد را مورد توجه قرار می‌دادند: قابلیت فهم، تطبیق‌پذیری و قابلیت حمل. در مقاله‌ی مذکور با تحلیل‌های تجربی صورت گرفته بر روی مؤلفه‌های Java Beans برای هر سنجه، آستانه‌هایی مشخص گردیده‌است. نویسندگان مقاله برای اندازه‌گیری قابلیت فهم کلی مؤلفه، سنجه‌های وجود فرا اطلاعات[۳۶] و نرخ قابلیت مشاهده‌ی مؤلفه[۳۷]، برای اندازه‌گیری میزان تطبیق‌پذیری، سنجه نرخ سفارشی‌سازی مؤلفه[۳۸] و در نهایت برای اندازه‌گیری قابلیت حمل، سنجه‌های خود-کمالی[۳۹] از منظر مقدار بازگشتی مؤلفه و خود-کمالی از منظر پارامترهای مؤلفه[۴۰] را پیشنهاد کرده‌اند. در ادمه مروری مختصر بر روی هریک از این سنجه‌ها خواهیم‌داشت.

زیر دسته ۱. ۱. ۲. ۱. ۱: سنجه وجود فرا اطلاعات

اگر فرا اطلاعاتی از یک مؤلفه ارائه شده‌باشد در این‌صورت کاربران آن مؤلفه به‌راحتی می‌توانند کاربرد آن‌را درک کنند. وجود فرا اطلاعات برطبق رابطه (۳۲) با کنترل وجود کلاس BeanInfo مشخص می‌شود.

$$EMI(c) = \begin{cases} 1 & beanInfo\ class\ exists \\ 0 & o.w \end{cases} \quad (۳۲)$$

این سنجه دارای دامنه‌ی اطمینان [۱ , ۰.۵] است.

زیر دسته ۱. ۱. ۲. ۱. ۲: سنجه نرخ قابلیت مشاهده‌ی مؤلفه

سنجه نرخ قابلیت مشاهده مؤلفه که طبق رابطه (۳۳)

قابل محاسبه می‌باشد، میزان سهولت در مشاهده‌ی یک مؤلفه را از نقطه نظر رفتار عملیاتی[۴۱]، پارامترهای ورودی پارامترهای خروجی نشان می‌دهد. رفتار یک مؤلفه‌ی جعبه سیاه می‌تواند عمدتا توسط متدهای خواندن[۴۲] که متناظر با مشخصات[۴۳] آن مؤلفه هستند مشخص شود.

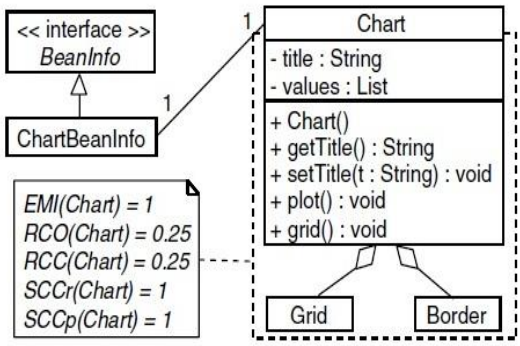
$$RCO(c) = \begin{cases} \frac{P_r(c)}{A(c)} & A(c) > ۰ \\ ۰ & o.w \end{cases} \quad (۳۳)$$

که در آن $P_r(c)$ تعداد مشخصات خواندنی در مؤلفه‌ی c $A(c)$ تعداد فیلدهای موجود در کلاس نمای[۴۴] مؤلفه‌ی c است.

این سنجه دارای دامنه‌ی اطمینان [۰ , ۰.۱۷ , ۰.۴۲] می‌باشد.

به‌عنوان مثال در شکل (۳) کلاس نمای Chart دارای ۴ فیلد (title، value، اشاره‌گر مرجع به نمونه‌ی Grid و اشاره‌گر مرجع به نمونه‌ی Broder) و یک متد خواندن (متناظر با فیلد title) است. بنابراین:

$$RCO(Chart) = \frac{۱}{۴} = ۰.۲۵$$



شکل (۳): نمودار کلاس

زیر دسته ۱. ۱. ۲. ۱. ۳: سنجه نرخ سفارشی‌سازی مؤلفه

منظور از قابلیت سفارشی‌سازی، قابلیت‌های پیش‌بینی شده برای پشتیبانی از سفارشی‌سازی و یا پیکربندی ویژگی‌های کارکردی یک مؤلفه است بر همین اساس، تعداد متدهای نوشتنی[۴۵] که یک مؤلفه ارائه می‌کند به شدت قابلیت سفارشی‌سازی آن‌را تحت تاثیر خود قرار می‌دهند. سنجه نرخ سفارشی‌سازی مؤلفه به‌وسیله‌ی رابطه (۳۴) قابل محاسبه است.

$$RCC(c) = \begin{cases} \frac{P_w(c)}{A(c)} & A(c) > ۰ \\ ۰ & o.w \end{cases} \quad (۳۴)$$

که در آن $P_r(c)$ تعداد مشخصات نوشتنی در مؤلفه‌ی c و $A(c)$ تعداد فیلدهای موجود در کلاس نمای مؤلفه‌ی c است.

این سنجه دارای دامنه‌ی اطمینان [۰ , ۰.۱۷ , ۰.۳۴] می‌باشد.

به‌عنوان مثال در شکل (۳) کلاس نمای Chart دارای ۴ فیلد (title، value، اشاره‌گر مرجع به نمونه‌ی Grid اشاره‌گر مرجع به نمونه‌ی Broder) و یک متد نوشتن (متناظر با فیلد title) است. بنابراین:

$$RCC(Chart) = \frac{۱}{۴} = ۰.۲۵$$

زیر دسته ۱. ۱. ۲. ۱. ۴: سنجه‌های وابستگی خارجی

وابستگی خارجی نشانگر درجه‌ی وابستگی یک مؤلفه به بقیه‌ی نرم‌افزاری است که از ابتدا آن مؤلفه را مورد استفاده قرار می‌دهد. فراخوانی متدهای کسب و کار[۴۶] مؤلفه از خارج آن، عملکردهایی را به بیرون ارائه می‌کند. بنابراین وابستگی خارجی متدهای کسب و کار پیاده‌سازی شده در داخل یک مؤلفه، وابستگی خارجی مؤلفه را تحت تاثیر خود قرار می‌دهند. پارامترها و یا مقادیر بازگشتی در متدهای کسب و کار ممکن است به بقیه‌ی نرم‌افزار بستگی داشته باشند. در مقابل، متدهای کسب و کاری که پارامتر و یا مقدار بازگشتی ندارند عملکردهایی را به خارج ارائه می‌کنند که به‌خودی خود کامل هستند[۴۷]. بر همین اساس در این قسمت دو سنجه را معرفی خواهیم‌کرد:

۱- سنجه خود-کمالی از منظر مقدار بازگشتی مؤلفه ($SCC_r(c)$): این سنجه برطبق رابطه (۳۵) قابل محاسبه می‌باشد.

$$SCC_r(c) = \begin{cases} \frac{B_v(c)}{B(c)} & B(c) > ۰ \\ ۱ & o.w \end{cases} \quad (۳۵)$$

که در آن $B_v(c)$ تعداد متدهای کسب و کار بدون مقدار بازگشتی و $B(c)$ تعداد کل متدهای کسب و کار در مؤلفه‌ی c است.

این سنجه دارای دامنه‌ی اطمینان [۱ , ۰.۶۱] می‌باشد.

به‌عنوان مثال در شکل (۳)، Chart دارای دو متد کسب و کار (grid و plot) است و نوع مقدار بازگشتی این متدها void می‌باشد. بنابراین:

$$SCC_r(Chart) = \frac{۲}{۲} = ۱$$

۲- سنجه خود-کمالی از منظر پارامتر مؤلفه ($SCC_p(c)$):

این سنجه برطبق رابطه (۳۶) قابل محاسبه می‌باشد.

$$SCC_p(c) = \begin{cases} \frac{B_p(c)}{B(c)} & B(c) > ۰ \\ ۱ & o.w \end{cases} \qquad (۳۶)$$

که در آن $B_p(c)$ تعداد متدهای کسب و کار بدون پارامتر و $B(c)$ تعداد کل متدهای کسب و کار در مؤلفه‌ی c است.

این سنجه دارای دامنه‌ی اطمینان [۰.۷۷ , ۰.۴۲] می‌باشد.

به عنوان مثال، در شکل (۳)، Chart دارای دو متد کسب و کار grid و plot است و این متدها هیچ پارامتری ندارند. بنابراین:

$$SCC_p(Chart) = \frac{۲}{۲} = ۱$$

زیر دسته ۱. ۱. ۲. ۲: سنجه‌های مبتنی بر واسط

در سال ۲۰۰۴، باگزال و همکارانش [۲۲] ادعا کردند که برای تعیین قابلیت استفاده‌ی مجدد، فهم واسط یک مؤلفه[۴۸] فاکتور کیفیت مهمی می‌باشد. آن‌ها برای اندازه‌گیری این فاکتور، مجموعه‌ای از سنجه‌ها شامل اندازه‌ی واسط[۴۹]، طول شناسه[۵۰] و تعداد آرگومان‌ها[۵۱] را معرفی کردند. در ادامه مروری مختصر بر روی هر یک از این سنجه‌ها خواهیم‌داشت.

زیر دسته ۱. ۱. ۲. ۲. ۱: سنجه اندازه‌ی واسط

این سنجه متوسط سایز اعلانات رویه[۵۲] از یک واسط را مشخص می‌کند و از طریق رابطه (۳۷) قابل محاسبه است.

$$\mathrm{APP} = \frac{\mathrm{n_a}}{\mathrm{n_p}} \qquad (۳۷)$$

که در آن $\mathrm{n_a}$ تعداد آرگومان‌ها و $\mathrm{n_p}$ تعداد رویه‌های فراخوانی شده توسط واسط می‌باشد.

واسط‌هایی با تعداد آرگومان‌های کمتر دارای قابلیت فهم بیشتری هستند در نتیجه هر چقدر APP کوچکتر باشد معیار قابلیت استفاده‌ی مجدد بزرگتر می‌شود.

زیر دسته ۱. ۱. ۲. ۲. ۲: سنجه تعداد و نرخ آرگومان‌های مجزا[۵۳]

سنجه تعداد آرگومان‌های مجزا، برای اندازه‌گیری میزان سازگاری در نام و نوع آرگومان‌ها ارائه شده‌است و برطبق رابطه (۳۸) قابل محاسبه می‌باشد.

$$DAC = |A| \qquad (۳۸)$$

که در آن نماد A نشانگر مجموعه‌ای از جفت نام-نوع‌هایی است که به عنوان آرگومان در واسط استفاده می‌شوند.

تذکر: در محاسبه‌ی این سنجه باید بر روی تعاریفی از متغیرها که از نظر نحوی متفاوت ولی از نظر معنایی یکسان هستند هم‌ارزی نحوی صورت داد و این تعاریف را در شمارش یکی حساب کرد.

سنجه دیگری که در این قسمت معرفی می‌کنیم نرخ آرگومان‌های مجزا است. این سنجه مستقل از سایز بوده و از طریق رابطه (۳۹) قابل محاسبه است.

$$DAR = \frac{DAC}{n_a} \qquad (۳۹)$$

که در آن n_a تعداد کل آرگومان‌های واسط است.

به عنوان مثال در مجموعه عبارات زیر، ۶ جفت مجزای نام-نوع و ۸ آرگومان وجود دارد:

$$void\ f1(int\ a, int\ b);$$
$$void\ f2(float\ a, int\ b);$$
$$void\ f3(float\ c, int\ b);$$
$$void\ f4(float\ *a, int\ *b);$$

بنابراین:

$$\therefore DAC = |A| = ۶.$$
$$\therefore DAR = \frac{DAC}{۸} = \frac{۶}{۸} = ۰.۷۵$$

زیر دسته ۱. ۱. ۲. ۲. ۳: سنجه مقیاس تکرار آرگومان[۵۴]

این سنجه نیز برای اندازه‌گیری میزان سازگاری بین نام‌گذاری و نوع آرگومان‌ها پیشنهاد شده و از طریق رابطه (۴۰) قابل محاسبه است.

$$ARS = \frac{\sum_{a \in A} |a|^2}{n_a} \quad (۴۰)$$

که در آن A یک مجموعه از نام-نوع‌ها، $|a|$ تعداد رویه-هایی که آرگومان a در آن استفاده شده‌است و n_a تعداد کل آرگومان‌های موجود در واسط می‌باشد.

زیر دسته ۱. ۱. ۲. ۲. ۴: سنجه متوسط اشتراک رشته‌ای[۵۵]

این سنجه معیاری برای اندازه‌گیری اشتراک یک مجموعه از شناسه‌ها می‌باشد و از طریق رابطه (۴۱) قابل محاسبه است.

$$MSC_A = \frac{\sum_{(x,y) \in comb(A)} \frac{|lcs(x,y)|}{\max(|x|,|y|)}}{n^{C_2}} \quad (۴۱)$$

که در آن A یک مجموعه از شناسه‌ها، n تعداد شناسه‌ها، $comb(A)$ مجموعه ترکیب‌های دو عنصر از مجموعه A و $lcs(x,y)$ طولانی‌ترین زیر دنباله‌ی مشترک x و y می-باشد.

اگر مجموعه‌ی شناسه‌ها دارای صفر و یا یک عضو باشد، MSC تعریف نشده خواهد بود.

زیر دسته ۱. ۱. ۲. ۲. ۵: سنجه میانگین طول شناسه

میانگین طول شناسه، میانگین وزن‌دار طول شناسه‌ها در واسط و میانه‌ی طول شناسه متوسط وزن‌دار طول شناسه‌ها می‌باشد. یکی از مشکلات میانگین طول شناسه این است که شناسه‌های غیرمعمول با طول بسیار زیاد و یا بسیار کم بر روی آن تاثیرگذار می‌باشد اما میانه‌ی طول شناسه این مشکل را برطرف کرده‌است.

زیر دسته ۱. ۱. ۲. ۲. ۶: سنجه چگالی آرگومان مرجع

این سنجه برای اندازه‌گیری رخداد آرگومان‌های مرجع در یک واسط پیشنهاد شده‌است و از طریق رابطه (۴۲) قابل محاسبه می‌باشد.

$$RAD = \frac{n_r}{n_a} \quad (۴۲)$$

که در آن n_r تعداد گذرهای آرگومان‌های مرجع و n_a تعداد کل آرگومان‌ها است.

هرچقدر مقادیر RAD بزرگتر باشد فهم واسط مشکل‌تر خواهدشد. در زبان C یک نوع آرگومان از طریق مرجع گذر داده می‌شود اگر شامل توکن * و در زبان C++ شامل توکن & باشد.

۳-۱-۳ زیر دسته ۱. ۱. ۳: رویکرد جعبه‌ی خاکستری

رویکرد جعبه‌ی خاکستری در استفاده‌ی مجدد مؤلفه، در واقع ترکیبی از رویکردهای جعبه‌ی سفید و جعبه‌ی سیاه می‌باشد. در ادامه و در این بخش به معرفی یکی از سنجه-های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد با این رویکرد خواهیم‌پرداخت.

زیر دسته ۱. ۱. ۳. ۱: سنجه‌های چیدمبر و کمرر

در سال ۱۹۹۴، چیدمبر و کمرر، رشته سنجه‌هایی را برای اندازه‌گیری نرم‌افزارهای شی‌گرا ارائه کردند [۱۷]. آن‌ها مدعی شدند که معیارهای پیشنهادی آن‌ها مستقل از نحو زبان می‌باشد و این سنجه‌ها می‌توانند درطول فازهای طراحی و پیاده‌سازی بر روی نمودار کلاس و یا کد منبع اعمال شوند. این سنجه‌ها که به سنجه‌های CK معروف هستند در جدول (۴) بیان شده‌اند.

۴-۱-۳ بررسی و تحلیل مزایا، محدودیت‌ها و اهداف سنجه‌های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد

در جدول (۵) به طور خلاصه اهداف، مزایا و محدودیت-های سنجه‌های عنوان شده برای اندازه‌گیری کمّی قابلیت استفاده مجدد مؤلفه بیان شده است. دسته مبتنی بر اندازه‌گیری ویژگی‌های اصلی ، سنجه‌های پایه برای محاسبه‌ی هزینه‌های استفاده مجدد، کارایی عملکردی و کیفیت مؤلفه‌ها را فراهم می‌آورند.

این گروه از سنجه‌ها علاوه بر سادگی در محاسبه، مستقل از زبان برنامه‌نویسی می‌باشند. بزرگترین ایراد این دسته، نیاز به داشتن کد منبع در محاسبه‌ی آن‌ها است. سنجه-های مبتنی بر معیارهای پیچیدگی و خاصیت پیمانه‌ای بودن نیز در دسته‌ی سنجه‌های جعبه‌ی سفید قرار دارند و برای محاسبه‌ی آن‌ها به کد منبع نیاز داریم. این سنجه‌ها با وجود این‌که به‌سادگی قابل محاسبه هستند اما مسئله‌ی

انتخاب وزن برای فاکتورهای موثر آن‌ها چالش برانگیز است.

سنجه‌های مبتنی بر عملکرد مؤلفه در دامنه، با درنظر گرفتن اینکه قابلیت استفاده‌ی مجدد به‌وسیله‌ی عملکردی که مؤلفه‌های نرم‌افزاری در دامنه‌شان دارند، مشخص می‌گردد به محاسبه‌ی قابلیت استفاده‌ی مجدد می‌پردازند. از طرفی در این دسته از سنجه‌ها ادعا شده‌است که قابلیت سفارشی‌سازی یک مولفه برروی قابلیت استفاده‌ی مجدد از آن تاثیر گذار است اما این ادعا به‌صورت ریاضی اثبات نشده‌است. در محاسبه‌ی این دسته از سنجه‌ها نیز نیاز به کد منبع داریم.

سنجه‌های مبتنی بر پیوستگی و وابستگی با معیار قرار دادن قابلیت اطمینان و تطبیق‌پذیری به اندازه‌گیری قابلیت استفاده‌ی مجدد مؤلفه‌ها از طریق محاسبه‌ی پیوستگی و وابستگی می‌پردازند. این سنجه‌ها در محاسبات خود روابط غیرمستقیم را نیز مد نظر قرار می‌دهند. وابسته به کد منبع بودن از جمله نکات منفی این دسته از سنجه‌ها محسوب می‌گردد.

سنجه‌های مبتنی بر قابلیت استفاده مجدد کلی مؤلفه، معیارهایی نظیر قابلیت فهم، تطبیق‌پذیری و قابلیت حمل را درنظر می‌گیرند. در این دسته، برای هر سنجه با تحلیل‌های تجربی آستانه‌هایی مشخص شده‌است البته لازم به تذکر است که این تحلیل‌های تجربی تنها بر روی جامعه‌ی محدودی از مولفه‌ها صورت پذیرفته‌است. عدم نیاز به کد مؤلفه از مزایای این دسته از سنجه‌هاست.

سنجه‌های مبتنی بر واسط با معیار قرار دادن میزان فهم واسط یک مؤلفه به اندازه‌گیری قابلیت استفاده‌ی مجدد از مؤلفه‌ها می‌پردازند. این دسته از سنجه‌ها نیازی به کد منبع ندارند و در دسته‌ی سنجه‌های جعبه‌ی سیاه می‌گنجند. برای این دسته از سنجه‌ها ارزیابی تجربی صورت گرفته‌است اما به‌صورت محدود، و برای مقایسه‌ی میزان قابلیت استفاده‌ی مؤلفه‌ها فقط دانش تخصصی نویسندگان معیار قرار گرفته‌است.

سنجه‌های CK با مدنظر قرار دادن پیچیدگی طراحی و مشخصه‌های اصلی نرم‌افزارهای شی‌گرا به ارزیابی طراحی‌های شی‌گرا می‌پردازند. از جمله مزایای این سنجه‌ها این است که مستقل از نحو زبان می‌باشند و می‌توانند درطول فازهای طراحی و پیاده‌سازی بر روی نمودار کلاس و یا کد منبع اعمال شوند. فقدان پایه‌ای تئوریک و یا فقدان ویژگی‌های ریاضیاتی مناسب در محاسبه‌ی سنجه‌ها از جمله نکات منفی این دسته بشمار می‌رود.

جدول (۴): سنجه‌های CK

نام سنجه	تعریف
CBO (وابستگی بین اشیاء)	دو کلاس را وابسته به هم گوییم اگر متدها یا متغیرهای نمونه در یک کلاس توسط دیگری استفاده شوند. در این صورت CBO برای یک کلاس تعداد کلاس‌هایی خواهد بود که با آن کلاس وابستگی دارند.
RFC (پاسخ کلاس)	تعداد کل متدهای کلاس به اضافه‌ی کل متدهای فراخوانی شده در دیگر کلاس‌ها
LCOM (فقدان همبستگی بین متدها)	تعداد جفت متدهای غیرمشابه در یک جفت کلاس
WMC (متدهای وزن‌دار در هر کلاس)	مجموع پیچیدگی همه‌ی متدهای یک کلاس
DIT (عمق درخت وراثت)	عمق درخت وراثت (بیشترین فاصله گره‌ها تا ریشه‌ی درخت)
NOC (تعداد فرزندان)	تعداد زیرکلاس‌های مستقیم از یک کلاس

جدول (۵): بررسی و تحلیل سنجه‌های ارائه‌شده برای اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد مؤلفه

پارامترهای ارزیابی / نام روش	رویکرد	ویژگی‌های استفاده‌شده	سنجه‌های پیشنهاد-شده	هدف	مزایا	محدودیت‌ها
سنجه‌های مبتنی بر اندازه‌گیری ویژگی‌های اصلی	جعبه‌ی سفید	هزینه‌های استفاده مجدد/ کارایی عملکردی/ کیفیت مؤلفه‌ها	حجم هالستید/ قائده-مندی/ بسامدِ استفاده‌ی مجدد/ پیچیدگی سایکلوماتیک مک-کِیب/	تشخیص و کیفیت-سنجی مولفه‌های قابل استفاده‌ی مجدد نرم‌افزاری	سادگی در محاسبه‌ی سنجه‌های پیشنهاد شده/ مستقل از زبان بودن	نیاز به کد منبع

ادامه جدول (۵)

پارامترهای ارزیابی / نام روش	رویکرد	ویژگی‌های استفاده‌شده	سنجه‌های پیشنهاد-شده	هدف	مزایا	محدودیت‌ها
سنجه‌های مبتنی بر معیارهای پیچیدگی و خاصیت پیمانه‌ای بودن	جعبه‌ی سفید	پیچیدگی / خاصیت پیمانه‌ای بودن	پیچیدگی داخلی کلاس/ پیچیدگی خارجی کلاس/ پیچیدگی همبستگی در یک کلاس/ پیچیدگی وابستگی در یک کلاس	اندازه‌گیری قابلیت استفاده‌ی مجدد و قابلیت نگهداری از نرم‌افزارهای شی‌گرا	سادگی در محاسبه‌ی سنجه‌های پیشنهاد شده	نیاز به کد منبع/ مسئله‌ی انتخاب وزن برای فاکتورهای موثر
سنجه‌های مبتنی بر عملکرد مؤلفه در دامنه	جعبه‌ی سفید	عملکردی که مؤلفه‌های نرم‌افزاری در دامنه‌شان دارند.	تنوع مؤلفه (برای اندازه-گیری قابلیت سفارشی-سازی)/ قابلیت استفاده‌ی مجدد از مؤلفه	اندازه‌گیری پیچیدگی، قابلیت سفارشی‌سازی، استفاده‌ی مجدد و قابلیت استفاده‌ی مجدد مؤلفه	ربط دادن قابلیت سفارشی‌سازی به قابلیت استفاده‌ی مجدد	نیاز به کد منبع/ رابطه‌ی بین قابلیت سفارشی‌سازی و قابلیت استفاده‌ی مجدد اثبات نشده است.
سنجه‌های مبتنی بر پیوستگی و وابستگی	جعبه‌ی سفید	قابلیت اطمینان و تطبیق‌پذیری	وابستگی/ همبستگی	اندازه‌گیری قابلیت استفاده‌ی مجدد مؤلفه‌های جاوا	علاوه بر روابط مستقیم، در نظرگرفتن روابط غیرمستقیم در محاسبه-ی پیوستگی و وابستگی	نیاز به کد منبع/ در ارزیابی سنجه‌ها تنها مؤلفه‌های جاوا مورد توجه قرار گرفته‌اند.
سنجه‌های مبتنی بر قابلیت استفاده مجدد کلی مؤلفه	جعبه‌ی سیاه	قابلیت فهم/ تطبیق‌پذیری/ قابلیت حمل	وجود فرا اطلاعات/ نرخ قابلیت مشاهده‌ی مؤلفه/ نرخ سفارشی‌سازی مؤلفه/ وابستگی خارجی	اندازه‌گیری قابلیت استفاده‌ی مجدد با تحلیل تجربی مؤلفه‌های Java Beans	برای هر سنجه با تحلیل‌های تجربی، آستانه‌هایی مشخص گردیده‌است/ عدم نیاز به کد مؤلفه	تحلیل‌های تجربی تنها بر روی مؤلفه‌های Java Beans صورت گرفته‌است.
سنجه‌های مبتنی بر واسط	جعبه‌ی سیاه	فهم واسط یک مؤلفه	اندازه‌ی واسط/ تعداد و نرخ آرگومان‌های مجزا/ مقیاس تکرار آرگومان/ متوسط اشتراک رشته‌ای/ میانگین طول شناسه/ چگالی آرگومان مرجع	تعیین سنجه‌هایی برای اندازه‌گیری قابلیت استفاده‌ی مجدد و ایجاد ابزاری ساده برای محاسبه‌ی خودکار آن‌ها	مورد توجه قرار دادن واسط مؤلفه برای اولین بار/ ایجاد ابزاری برای ارزیابی تجربی و محاسبه‌ی خودکار سنجه‌ها/ عدم نیاز به کد مؤلفه	برای ارزیابی تجربی تنها دوازده مؤلفه به زبان‌های C و C++ انتخاب شده‌اند/ برای مقایسه‌ی میزان قابلیت استفاده‌ی مؤلفه‌ها فقط دانش تخصصی نویسندگان معیار قرار گرفته است

سنجه‌های CK	جعبه‌ی خاکستری	پیچیدگی طراحی/ مشخصه‌های اصلی نرم‌افزارهای شی‌گرا	عمق درخت وراثت/ وابستگی بین اشیاء/ تعداد فرزندان/ متدهای وزن‌دار در هر کلاس/ فقدان همبستگی بین متدها/ پاسخ کلاس	ارزیابی طراحی‌های شی‌گرا	مستقل از نحو زبان بودن	نیاز به کد منبع/ فقدان پایه‌ای تئوریک/ فقدان ویژگی‌های ریاضیاتی مناسب

۲-۳- دسته ۱. ۲: مدل‌های ارزیابی قابلیت استفاده‌ی مجدد

مدل بیان‌کننده‌ی ارتباط بین سنجه‌ها و یکی از روش‌های ارزیابی کمّی کیفیت نرم‌افزار است. در این بخش مروری بر مدل‌های ارائه شده در مورد قابلیت استفاده‌ی مجدد انجام داده‌ایم و مانند بخش قبل، آن‌ها را به سه دسته‌ی کلی تقسیم کرده‌ایم: ۱- مدل‌های جعبه سفید، ۲- مدل‌های جعبه سیاه و ۳- مدل‌های جعبه خاکستری. در هر سه دسته‌ی ارائه شده با هدف نشان دادن میزان پیشرفت در اندازه‌گیری قابلیت استفاده‌ی مجدد در طی زمان، نتایج تحقیقات قبلی را به ترتیب زمانی پشت سرهم آورده‌ایم.

۱-۲-۳ زیر دسته ۱. ۲. ۱: رویکرد جعبه‌ی سفید

همانطور که قبلا نیز اشاره کردیم در رویکرد جعبه‌ی سفید، کد منبع مؤلفه در اختیار توسعه‌دهنده بوده و قبل از این‌که در متن جدیدی به‌کار گرفته‌شود می‌تواند تغییر یابد. در ادامه و در این بخش، تعدادی از مهمترین مدل-های ارائه شده در رابطه با اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد از مؤلفه با این رویکرد را عنوان خواهیم‌کرد.

زیر دسته ۱. ۲. ۱. ۱: مدل ارزیابی تجربی مبتنی بر کاویدن منابع نرم‌افزاری

در سال ۲۰۰۵، سلبای با استخراج منابع از یک محیط توسعه‌ی نرم‌افزار در ناسا که مرتبا نرم‌افزاری را مورد استفاده‌ی مجدد قرار می‌دهد به بررسی، تجزیه و تحلیل قابلیت استفاده‌ی مجدد پرداخت [۲۴]. او دو دسته فاکتور به نام فاکتورهای طراحی ماژول و فاکتورهای پیاده‌سازی ماژول را تعریف کرد و چهار کلاس از ماژول‌های نرم‌افزاری را مورد تحلیل قرار داد: ماژول‌هایی که بدون بازبینی مورد استفاده‌ی مجدد قرار می‌گیرند، ماژول‌هایی که با بازبینی جزئی مورد استفاده‌ی مجدد قرار می‌گیرند (کمتر از ۲۵ درصد بازبینی)، ماژول‌هایی که با بازبینی کلی مورد استفاده‌ی مجدد قرار می‌گیرند (بیشتر از ۲۵ درصد بازبینی) و ماژول‌هایی که تازه توسعه‌یافته‌اند. همچنین در این پژوهش نرخ شکست ماژول‌هایی که مورد استفاده‌ی مجدد و یا تغییر قرار گرفته‌اند و یا اینکه به‌تازگی توسعه پیدا کرده‌اند، مورد بررسی قرار گرفته‌است. در نهایت سلبای به این نتیجه رسید که ماژول‌هایی که بدون بازبینی مورد استفاده‌ی مجدد قرار می‌گیرند خطاهای کمتری دارند. در مقابل ماژول‌هایی که به‌صورت کلی مورد بازبینی قرار می‌گیرند بیشتر مستعد خطا هستند.

زیر دسته ۱. ۲. ۱. ۲: مدل مبتنی بر همبستگی و وابستگی

در سال ۲۰۰۸، گوی و همکارانش [۲۰] برای اندازه‌گیری قابلیت استفاده‌ی مجدد از مؤلفه‌ها معیارهای جدیدی را برای وابستگی و همبستگی ارائه‌کردند و نشان دادند که این معیارها جزء برترین معیارهای ارائه‌شده در رابطه با اندازه‌گیری قابلیت استفاده‌ی مجدد از مؤلفه هستند. WTCoup، CF، CBO، RFC و DAC سنجه‌های وابستگی و WTCoh، RLCOM، LCOM، LCOM۳ و TCC سنجه‌های پیوستگی استفاده‌شده توسط آن‌ها می-باشند.

زیر دسته ۱. ۲. ۱. ۳: مدل مبتنی بر الگوریتم‌های شبکه عصبی

در سال ۲۰۱۰، منهاس و همکارانش برای ارزیابی قابلیت استفاده‌ی مجدد از مؤلفه‌های نرم‌افزاری مدل ارزیابی قابلیت استفاده‌ی مجددی را ارائه کردند [۲۸]. در واقع

آن‌ها با آزمایش پنج رویکرد مختلف مبتنی بر شبکه‌ی عصبی و با درنظرگرفتن مقادیر سنجه‌ها به‌عنوان ورودی، مشخصه‌های ساختاری و سنجه‌هایی نرم‌افزاری را برای ارزیابی قابلیت استفاده‌ی مجدد از مؤلفه‌ها ارائه کردند. در این مطالعه مقدار قابلیت استفاده‌ی مجدد با هدف شناسایی خودکار مؤلفه‌های با کیفیت محاسبه شده‌است و سنجه‌های پیشنهاد شده در این راستا عبارتند از: پیچیدگی سایکلوماتیک مک-کِیب، شاخص هالستید، سنجه‌ی قائده‌مندی، سنجه‌ی بسامد استفاده‌ی مجدد و سنجه‌ی وابستگی.

زیر دسته ۱. ۲. ۱. ۴: مدل دسته‌بندی قابلیت استفاده‌ی مجدد

در سال ۲۰۱۱، کومار و همکارانش برای دسته‌بندی قابلیت استفاده‌ی مجدد از مؤلفه‌های نرم‌افزاری مدلی را ارائه کردند که در آن از طبقه‌بند SVM استفاده شده‌بود [۳۰]. در این مطالعه سنجه‌هایی که برای شناسایی ماژول‌های نرم‌افزاری قابل استفاده‌ی مجدد استفاده شده-اند عبارتند از: پیچیدگی سایکلوماتیک مک-کِیب، شاخص هالستید، سنجه‌ی قائده‌مندی، سنجه‌ی بسامد استفاده‌ی مجدد و سنجه‌ی وابستگی.

۲-۲-۳ زیر دسته ۱. ۲. ۲: رویکرد جعبه‌ی سیاه

همانطور که قبلا نیز اشاره کردیم در رویکرد جعبه‌ی سیاه، تنها واسط مؤلفه (شامل متدها و صفت‌های عمومی) و مستندات آن قابل مشاهده هستند. در ادامه و در این بخش، تعدادی از مهمترین مدل‌های ارائه شده در رابطه با اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد از مؤلفه با این رویکرد را عنوان خواهیم‌کرد.

زیر دسته ۱. ۲. ۱: مدل مبتنی بر آزمایشات تجربی

در سال ۱۹۹۸ بارنارد برای استفاده مجدد از نرم افزارهای شی‌گرا مدلی را که از دو آزمایش تجربی بدست آمده بود پیشنهاد کرد [۳۲]. این‌بار نیز انواع سنجه‌های نرم افزاری در دسترس به عنوان پایه و اساس مدل مذکور مورد استفاده قرار داده‌شدند. البته بر اساس آزمایشات صورت گرفته، سنجه‌هایی انتخاب می‌شدند که به بهترین وجه ممکن به استفاده‌ی مجدد مربوط باشند. در این آزمایش‌ها کلاس‌هایی از C++، Java و کتابخانه‌های Eiffel (با فرض اینکه کلاس‌های موجود در کتابخانه‌ها قابلیت استفاده‌ی مجدد بیشتری دارند) شرکت داشتند. مجموعه سنجه‌های بارنارد بر سادگی، عمومیت و قابلیت فهم واسط‌ها، متدها و صفت‌های کلاس تمرکز دارند.

زیر دسته ۱. ۲. ۲. ۲: مدل قابلیت استفاده‌ی مجدد کلی مؤلفه

در سال ۲۰۰۳، واشیزاکی و همکارانش با درنظر گرفتن معماری JavaBeans مدلی را برای اندازه‌گیری قابلیت استفاده‌ی مجدد از مؤلفه‌های جعبه‌ی سیاه پیشنهاد کردند [۵]. این مدل که بر طبق رابطه (۴۳) قابل محاسبه است، ترکیبی از سنجه‌های EMI، SCCr و RCC می-باشد و در واقع قابلیت فهم، تطبیق‌پذیری و قابلیت حمل یک مؤلفه را مورد ارزیابی قرار می‌دهد.

$$COR(c) = ۱.۷۶ * \frac{V_{EMI}(c)+V_{RCC}(c)+V_{SCCr}(c)}{۳} - ۱.۱۳ \quad (۴۳)$$

اگر مقدار $COR(c)$ بزرگتر از صفر باشد آنگاه می‌توان گفت که سطح قابلیت استفاده‌ی مجدد مؤلفه c بالا است.

زیر دسته ۱. ۲. ۳: مدل مبتنی بر مقدار اندازه‌گیری-شده قابلیت استفاده‌ی مجدد

در سال ۲۰۱۲، ینگ-می و همکارانش مدلی را برای ارزیابی قابلیت استفاده‌ی مجدد ارائه دادند. در این مدل که از طریق رابطه (۴۴) قابل محاسبه است، قابلیت استفاده‌ی مجدد به‌عنوان فاکتوری وابسته به کارکرد[۵۶]، قابلیت اطمینان[۵۷]، قابلیت استفاده[۵۸]، قابلیت نگهداری و قابلیت حمل[۵۹] در نظرگرفته شده‌است [۳۳].

$$RMV = W_۱ * F + W_۲ * R + W_۳ * U + W_۴ * M + W_۵ * P \quad (۴۴)$$

که در آن $W_i(i = ۱, \dots, ۵)$ وزن، F کارکرد، R قابلیت اطمینان، U قابلیت استفاده، M قابلیت نگهداری و P قابلیت حمل می‌باشد.

۳-۲-۳ زیر دسته ۱. ۲. ۳: رویکرد جعبه‌ی خاکستری

همانطور که قبلا نیز اشاره کردیم رویکرد جعبه‌ی خاکستری ترکیبی از رویکردهای جعبه‌ی سفید و جعبه‌ی سیاه می‌باشد. در ادامه و در این بخش، تعدادی از مهمترین مدل‌های ارائه شده در رابطه با اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد از مؤلفه با این رویکرد را عنوان خواهیم‌کرد.

زیر دسته ۱. ۲. ۳. ۱: مدل تحلیل‌گر خودکار

در سال ۲۰۰۱ اتزکورن و همکارانش مدلی را ارائه کردند که قابلیت استفاده‌ی مجدد از نرم‌افزارهای شی‌گرا را اندازه می‌گرفت [۲۳]. در مقاله‌ی آن‌ها مجموعه‌ی جامعی از متریک‌ها پیشنهاد شد که جنبه‌های مختلف از قابلیت استفاده‌ی مجدد کلاس‌ها را تحت پوشش خود قرار می‌داد و به‌صورت مجموع متریک‌های مربوط به قابلیت پیمانه‌ای بودن، سایز واسط، مستندسازی و پیچیدگی یک کلاس تعریف شده‌بود. لازم به‌ذکر است که در این مجموع به همه‌ی متریک‌ها وزن یکسانی نسبت داده شده‌است.

زیر دسته ۱. ۲. ۳. ۲: مدل مبتنی بر رویکرد فازی-عصبی

در سال ۲۰۰۶، ساندهو و همکارانش برای اندازه‌گیری قابلیت استفاده‌ی مجدد از مؤلفه‌های نرم‌افزاری مدلی را پیشنهاد کردند [۲۵]. آن‌ها سنجه‌های CK را مورد تحلیل قرار داده و ناسازگاری‌های موجود را حذف کردند و در گام بعد، به طراحی چارچوبی برای ارزیابی قابلیت استفاده‌ی مجدد پرداختند. در مدل ارائه‌شده توسط آن‌ها، برای ارزیابی قابلیت استفاده‌ی مجدد، موتوری استنتاجی استفاده شده‌است که به‌صورت عصبی-فازی کار می‌کند و در آن ورودی‌هایی در شکل ویژگی‌های ساختاری به سیستم وارد شده و خروجی‌ای برحسب قابلیت استفاده‌ی مجدد تولید می‌شود. رشته سنجه‌هایی که در این مدل مورد استفاده قرار گرفته‌اند به این شرح می‌باشند: TWMC[۶۰]، LTDIT[۶۱]، LTNOC[۶۲]، LCBO[۶۳] و LCOM[۶۴] (این سنجه‌ها همان سنجه‌های CK هستند با این تفاوت که مقادیر آن‌ها بین صفر تا یک محدود شده‌است).

زیر دسته ۱. ۲. ۳. ۳: مدل مبتنی بر رویکرد Taguchi

در سال ۲۰۰۷، ساندهو و همکارانش درمورد تاثیر فاکتورهای سهیم در اندازه‌گیری قابلیت استفاده‌ی مجدد از مؤلفه‌های نرم‌افزاری بررسی کمّی‌ای را انجام دادند که به ارزیابی کیفیت مؤلفه‌ها کمک می‌کرد [۲۶]. آن‌ها در تحلیل اهمیت ویژگی‌های مختلف، رویکرد Taguchi را استفاده کردند. از نتایج بدست آمده اینگونه استنباط می‌شود که در تصمیم‌گیری در مورد قابلیت استفاده‌ی مجدد از یک مؤلفه‌ی نرم‌افزاری تابع‌گرا مهمترین فاکتور، پیچیدگی می‌باشد و در حالت شی‌گرا، وابستگی و پیچیدگی توامان باهم نقش مهمی را در ایجاد سطح بالای قابلیت استفاده‌ی مجدد ایفا می‌کنند. در مورد الگوی تابع‌گرا سنجه‌های ارائه شده در این مطالعه عبارتند از: پیچیدگی سایکلوماتیک مک-کِیب، شاخص هالستید، سنجه‌ی قائده‌مندی، سنجه‌ی بسامد استفاده‌ی مجدد و سنجه‌ی وابستگی و در مورد الگوی شی‌گرا سنجه‌های ارائه شده عبارتند از: TWMC، LTDIT، LTNOC، LCBO و LCOM.

زیر دسته ۱. ۲. ۳. ۴: مدل قابلیت استفاده‌ی مجدد در سیستم‌های شی‌گرا

در سال ۲۰۰۹، ساندهو و همکارانش سیستم ارزیابی قابلیت استفاده‌ی مجددی را برای مؤلفه‌های نرم‌افزاری شی‌گرا ارائه کردند [۲۷]. آن‌ها سنجه‌هایی را پیشنهاد نمودند که از رویکردهای مختلف، مبتنی بر شبکه‌های عصبی به‌دست آمده‌بود. لازم به‌ذکر است که تاکنون رویکردهای شبکه‌ی عصبی مختلفی برای مدل‌سازی داده‌های قابلیت استفاده‌ی مجدد ارائه شده‌است که در مقاله‌ی مذکور، چهارده عدد از این الگوریتم‌های شبکه‌ی عصبی مورد آزمایش قرار گرفته‌اند.

زیر دسته ۱. ۲. ۳. ۵: مدل ویژگی قابلیت استفاده‌ی مجدد

در سال ۲۰۱۱، فیضل-امین و همکارانش برای اندازه‌گیری قابلیت استفاده‌ی مجدد از مؤلفه‌های نرم‌افزاری مدل قابلیت استفاده‌ی مجددی را ارائه کردند [۲۹]. در این راستا آن‌ها شش مشخصه‌ی مرتبط با قابلیت استفاده‌ی مجدد از مؤلفه‌های نرم‌افزاری را پیشنهاد نمودند. مدل ارائه‌شده با استفاده از رویکرد GQM[۶۵] به‌دست آمده‌بود و به فهم فاکتورها برای اندازه‌گیری کیفیت نرم‌افزار کمک می‌کرد.

علاوه‌براین آن‌ها سنجه‌هایی را نیز ارائه کردند که برای اندازه‌گیری قابلیت استفاده‌ی مجدد استفاده می‌شد. این سنجه‌ها عبارتند از: سنجه‌ی اندازه، سنجه‌ی وابستگی، سنجه‌ی همبستگی و سنجه‌ی تغییرپذیری. در این مطالعه سنجه‌های استفاده شده برای اندازه‌گیری اندازه، LOC و NOM، سنجه‌ی استفاده شده برای اندازه‌گیری وابستگی، CBO و سنجه‌ی استفاده شده برای اندازه‌گیری همبستگی LCOM هستند. همچنین سنجه‌های تغییرپذیری بر اساس تئوری و مکانیزم وراثت بوده و به‌ترتیب عبارتند از عمق درخت وراثت و تعداد فرزندان.

زیر دسته ۱. ۲. ۳. ۶: مدل مبتنی بر سازمان‌دهی مشخصه‌ها

در سال ۲۰۱۲ هریستو و همکارانش مدلی را برای اندازه-گیری قابلیت استفاده‌ی مجدد ارائه کردند [۳۱]. آن‌ها در بیان هدف اصلی مدل خود عنوان کرده‌اند که به دنبال کشف مشخصه‌های جدیدی از مؤلفه‌های نرم‌افزاری که بیان‌کننده‌ی قابلیت استفاده‌ی مجدد آن‌ها باشد، نبوده‌اند بلکه هدف اصلی آن‌ها سازمان‌دهی کردن بهتر مشخصه-های موجود و تعیین مجموعه‌ای مرجع از همین مشخصه-های شناخته شده‌است. هسته‌ی اصلی مدل اندازه‌گیری ارائه شده توسط آن‌ها براساس این مشخصه‌ها شکل گرفته است: قابلیت دسترس‌پذیری، مستندسازی، پیچیدگی، کیفیت، قابلیت نگهداری، قابلیت تطبیق‌پذیری، استفاده‌ی مجدد و قیمت.

۴-۲-۳ بررسی و تحلیل مدل‌های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد

در جدول (۶) رویکرد، روش و ویژگی‌های استفاده‌شده در مدل‌های عنوان گردیده در این مقاله برای اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد مؤلفه بیان شده‌اند.

جدول (۶): بررسی و تحلیل مدل‌های ارائه‌شده برای اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد مؤلفه

پارامترهای ارزیابی / نام مدل	رویکرد	روش	ویژگی‌های استفاده شده	سنجه‌های استفاده‌شده
تحلیلگر خودکار	جعبه‌ی خاکستری	رویکرد مبتنی بر سنجه	قابلیت پیمانه‌ای بودن، سایز واسط، مستندسازی و پیچیدگی	فقدان همبستگی بین متدها، تعداد تنظیم‌شده‌ی پیغامهایی که فرستاده می‌شوند[۶۶]، تعداد تنظیم-شده‌ی توابع دوست[۶۷]، تعداد تنظیم‌شده‌ی دسترسی به متغیرهای خارجی[۶۸]، تعداد تنظیم-شده‌ی توابع عمومی[۶۹]، تعداد خطوط توضیحات متد، تعداد خطوط کد منبع

ارزیابی تجربی مبتنی بر کاویدن منابع نرم‌افزاری	جعبه‌ی سفید	رویکرد مبتنی بر روش (GQM)	فاکتورهای طراحی ماژول، فاکتورهای پیاده‌سازی ماژول، فاکتورهای شکست ماژول	درصد ماژول‌هایی که بدون بازبینی مورد استفاده‌ی مجدد قرار گرفته‌اند، درصد ماژول‌هایی که با بازبینی جزئی مورد استفاده‌ی مجدد قرار گرفته‌اند، درصد ماژول‌هایی که با بازبینی کلی مورد استفاده‌ی مجدد قرار گرفته‌اند، تلاش لازم برای توسعه‌ی ماژول، نرخ شکست ماژول، تعداد خطوط کد منبع، پیچیدگی سایکلوماتیک، نرخ تصحیح خطا

ادامه جدول (۶)

پارامترهای ارزیابی / نام مدل	رویکرد	روش	ویژگی‌های استفاده شده	سنجه‌های استفاده‌شده
مدل مبتنی بر رویکرد عصبی- فازی	جعبه خاکستری	رویکرد عصبی - فازی		LCBO ،LTNOC ،LTDIT ،TWMC و LCOM
مدل مبتنی بر رویکرد Taguchi	جعبه خاکستری	رویکرد Taguchi		پیچیدگی سایکلوماتیک، حجم هالستید، قائده-مندی، بسامد استفاده‌ی مجدد، وابستگی، LCBO ،LTNOC ،LTDIT ،TWMC و LCOM
مدل مبتنی بر پیوستگی و وابستگی	جعبه‌ی سفید	رویکرد مبتنی بر رگرسیون خطی و ضریب همبستگی رتبه	معیارهای جدید همبستگی و وابستگی بدست آمده	DAC ،RFC ،CBO ،CF ،WTCoup، WTCoh ،LCOM ،RLCOM ،LCOM۳ و TCC
مدل قابلیت استفاده‌ی مجدد در سیستم‌های شی‌گرا	جعبه‌ی خاکستری	رویکرد مبتنی بر شبکه‌ی عصبی		LCBO ،LTNOC ،LTDIT ،TWMC و LCOM
مدل مبتنی بر الگوریتم-های شبکه‌ی عصبی	جعبه‌ی سفید	رویکرد مبتنی بر شبکه‌ی عصبی		پیچیدگی سایکلوماتیک، حجم هالستید، قائده-مندی، بسامد استفاده‌ی مجدد، وابستگی
مدل ویژگی قابلیت استفاده‌ی مجدد	جعبه‌ی خاکستری	رویکرد مبتنی بر روش (GQM)	قابلیت نگهداری، قابلیت حمل، قابلیت انعطاف‌پذیری، قابلیت فهم، پوشش حوزه، استقلال	تعداد خطوط کد منبع، CBO، LCOM، DIT، NOM و NOC
مدل دسته‌بندی قابلیت استفاده‌ی مجدد	جعبه‌ی سفید	رویکرد مبتنی بر طبقه‌بند SVM		پیچیدگی سایکلوماتیک، حجم هالستید، قائده-مندی، بسامد استفاده‌ی مجدد، وابستگی
مدل مبتنی بر سازمان-دهی مشخصه‌ها	جعبه خاکستری	رویکرد مبتنی بر روش (GQM)	قابلیت دسترس‌پذیری، مستندسازی، پیچیدگی، کیفیت، قابلیت نگهداری، قابلیت تطبیق-پذیری، استفاده‌ی مجدد و قیمت	پیچیدگی سایکلوماتیک، تعداد خطوط کد منبع، وابستگی، پیوستگی، نرخ سفارشی‌سازی مؤلفه و ...

مدل مبتنی بر آزمایشات تجربی	جعبه‌ی سیاه	رویکرد مبتنی بر سنجه	سادگی، عمومیت، قابلیت فهم واسط‌ها، متدها و صفت‌های کلاس	وابستگی بین اشیاء، پیچیدگی تصحیح متدها و پیچیدگی واسط متد
مدل قابلیت استفاده‌ی مجدد کلی مؤلفه	جعبه‌ی سیاه	رویکرد مبتنی بر سنجه	قابلیت فهم، تطبیق‌پذیری، قابلیت حمل	وجود فرا اطلاعات، نرخ قابلیت مشاهده‌ی مؤلفه، نرخ سفارشی‌سازی مؤلفه، وابستگی خارجی
مدل مبتنی بر مقدار اندازه‌گیری شده‌ی قابلیت استفاده‌ی مجدد	جعبه‌ی سیاه	رویکرد مبتنی بر سنجه+ رویکردی جدید با توجه به اطلاعات بازگشتی کاربر	کارکرد، قابلیت اطمینان، قابلیت استفاده، قابلیت نگهداری و قابلیت حمل	نرخ استفاده‌ی مجدد، تعداد دفعات استفاده‌ی مجدد، نرخ استفاده‌ی مجدد کد، نرخ استفاده‌ی مجدد تابع، پیچیدگی واسط

۴- مقایسه‌ی مرور انجام‌شده با دیگر مرورهای موجود

همانطورکه قبلا نیز اشاره کردیم در زمینه‌ی اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد مؤلفه‌های نرم‌افزاری تاکنون مرورهایی صورت پذیرفته است اما هیچ‌کدام از آن‌ها از جامعیت لازم برخوردار نیستند. در این بخش قصد داریم به تحلیل این مرورها و مقایسه‌ی آن‌ها با مرور انجام گرفته در این پژوهش بپردازیم.

با جستجوهایی که انجام دادیم به ۸ مقاله‌ی مروری در زمینه‌ی اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد از مؤلفه‌های نرم‌افزاری رسیدیم [۳۵-۴۲]. در این میان تنها در مقالات [۳۶]، [۴۰] و [۴۲]، مرور به روشی نظامند صورت پذیرفته است. از طرفی تا آنجایی که ما می‌دانیم تاکنون در هیچ مروری سعی نشده‌است تا به‌طور کامل سنجه‌ها و مدل‌های ارزیابی کمّی قابلیت استفاده‌ی مجدد، در کنار هم مورد بررسی قرار بگیرند و در اکثر مرورهای صورت گرفته، فقط روش‌های موجود در یکی از این دسته‌ها مورد مطالعه قرار گرفته‌اند. به طور مثال در مقالات مروری [۳۵]،[۳۷]، [۴۱] و [۴۲]، تنها برخی از سنجه‌های اندازه‌گیری قابلیت استفاده‌ی مجدد از مؤلفه‌های نرم‌افزاری مورد مطالعه قرار گرفته‌اند و در مقالات مروری [۳۸] و [۴۰] تنها به مطالعه‌ی تعداد محدودی از مدل‌های ارائه‌شده در زمینه‌ی اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد پرداخته شده‌است. البته تعدادی مطالعه‌ی مروری مانند [۳۶] و [۳۹] وجود دارند که به بررسی روش‌های محدودی از هر دو دسته‌ی مدل و سنجه پرداخته‌اند اما در هیچ‌کدام از آن‌ها، بین مدل و سنجه تفاوتی قائل نشده‌است. نکته‌ی دیگری که در اینجا قابل توجه می‌باشد این است که در هیچ‌کدام از این مقالات مروری به جز مقاله [۳۶] سعی نشده‌است تا روش‌های ارائه شده دسته‌بندی شوند. در مقاله‌ی مروری [۳۶] روش‌های موجود به چند طریق دسته‌بندی شده‌اند اما برخی از این دسته‌بندی‌ها از جامعیت لازم برخوردار نیستند به طوری که تعدادی از روش‌ها در هیچ زیر دسته‌ای قرار نمی‌گیرند. همچنین تنها در مقالات [۳۸] و [۴۰] بین روش‌های ارائه شده مقایسه صورت گرفته‌است. در جدول (۷) به‌صورت خلاصه مقایسه‌ای بین مرور جاری و دیگر مرورهای موجود در حوزه‌ی اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد از مؤلفه‌ها انجام گرفته‌است.

جدول (۷): مقایسه‌ی مرور جاری با دیگر مرورهای موجود در حوزه‌ی اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد از مؤلفه‌ها

پارامترهای ارزیابی / مرجع	نوع مرور	تعداد مقالات پایه	بررسی مدل‌های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد	بررسی سنجه‌های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد	دسته‌بندی روش‌های بررسی‌شده	مقایسه‌ی روش‌های بررسی‌شده

پژوهش جاری ۲۰۱۵	نظام‌مند	۲۱	*	*	*	*
[۴۲] ۲۰۱۴	نظام‌مند	۱۹		*		
[۴۱] ۲۰۱۳	غیر نظام‌مند	۱۴		*		
[۳۸] ۲۰۱۳	غیر نظام‌مند	۳	*			*

ادامه جدول (۷)

پارامترهای ارزیابی / مرجع	**نوع مرور**	**تعداد مقالات پایه**	**بررسی مدل‌های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد**	**بررسی سنجه‌های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد**	**دسته‌بندی روش‌های بررسی‌شده**	**مقایسه‌ی روش‌های بررسی‌شده**
[۴۰] ۲۰۱۲	نظام‌مند	۹	*			*
[۳۶] ۲۰۱۱	نظام‌مند	۲۰	*	*	*	
[۳۹] ۲۰۱۰	غیر نظام‌مند	زیر ۱۰	*	*		
[۳۷] ۲۰۰۸	غیر نظام‌مند	زیر ۱۰		*		
[۳۵] ۲۰۰۷	غیر نظام‌مند	۴		*		

۵- چالش‌ها و مسائل باز موجود

آشکارترین نقطه ضعفِ سنجه‌ها و مدل‌هایی که تا اینجا آن‌ها را مرور کرده‌ایم این است که تقریبا همه‌ی آن‌ها فاقد اعتبارسنجی تجربی مناسب، برای قابلیت‌هایی که برایشان ادعا شده‌است می‌باشند و سرچشمه‌ی اکثر آن‌ها به‌طور عمده نظرات کارشناسان و یا شواهد به‌دست‌آمده از مطالعات موردی کوچک است. بنابراین به‌نظر می‌رسد که تاکنون تحقیقات انجام شده برروی قابلیت استفاده‌ی مجدد در پژوهش‌های تجربی مهندسی نرم‌افزار، کمتر نمود پیدا کرده‌است. البته برای این وضعیت دلایل زیادی وجود دارد: یک دلیل محکم این است که در جامعه‌ی تحقیقاتی، هیچ توافقی برروی اینکه کدام مشخصه‌های نرم‌افزاری، پایه‌ای مناسب را برای تعیین قابلیت استفاده‌ی مجدد از نرم‌افزار ارائه می‌کنند و یا اینکه کدام سنجه‌ها برای اندازه‌گیری این مشخصه‌ها کافی هستند، وجود ندارد. به‌عبارت دیگر هنوز هیچ درک مشترکی از اینکه قابلیت استفاده‌ی مجدد از نرم‌افزار چه هست و چگونه می‌توان آن را اندازه گرفت وجود ندارد. علاوه براین همان‌طور که می‌توان دید، در مجموعه سنجه‌ها و مدل-هایی که ارائه کرده‌ایم تمرکز اصلی بر روی ویژگی‌های فنی نرم‌افزار است که می‌توانند با انتظارات متخصصین مغایرت داشته باشند. بنابراین برای تعریف و اندازه‌گیری

قابلیت استفاده‌ی مجدد، رویکرد جامع‌تری مورد نیاز می‌باشد. با جستجو در منابع می‌توان دید که دیدگاه GQM می‌تواند رویکردی مناسب در ارائه‌ی مدل قابلیت استفاده‌ی مجددی ساخت‌یافته‌تر، تحلیلی‌تر و عادلانه‌تر باشد [۳۲، ۳۴]. این مدل به ما کمک خواهدکرد تا درک بهتری از قابلیت استفاده‌ی مجدد از نرم‌افزار داشته باشیم و این باعث خواهدشد که از یک طرف محققین تشویق شوند تا تلاش بیشتری را برای اعتبارسنجی تجربی مدل‌ها صرف کنند و از طرف دیگر به متخصصین اعتماد لازم برای اندازه‌گیری قابلیت استفاده‌ی مجدد در دنیای واقعی را خواهدداد.

۶- نتیجه‌گیری و کارهای آینده

مطالعه‌ی انجام شده در این مقاله مروری نظام‌مند بر ادبیات گذشته در مورد روش‌های اندازه‌گیری کمّی قابلیت استفاده‌ی مجدد از مؤلفه است. در این مقاله روش‌های موجود به دو گروه عمده‌ی سنجه‌ها و مدل‌ها تقسیم شده‌اند و گزارشی جامع، کامل و عادلانه در مورد این روش‌ها ارائه گردیده‌است. از آنجایی که میزان سودمندی و عملی بودن مدل‌ها و سنجه‌ها را تنها می‌توان با انجام مطالعات موردیِ تجربی (که به اندازه‌ی کافی بزرگ و جامع باشند) و یا آزمون‌های آماری بر روی داده‌های واقعی از کتابخانه‌های موجود (مربوط به قطعات قابل استفاده‌ی مجدد) اثبات کرد به‌وضوح می‌توان دید که اشکال عمده‌ی روش‌های اندازه‌گیری پیشنهاد شده این است که فاقد اعتبار سنجی تجربی مناسب می‌باشند. بنابراین به‌صورت منطقی گام بعدی این خواهدبود که برپایه‌ی این سنجه‌ها و با دنبال کردن رویکردهایی مناسب مثل (GQM) مدلی مناسب برای قابلیت استفاده‌ی مجدد را پیاده‌سازی کرد و این مدل محاسباتی را تا زمانیکه نیازمندی‌های متخصصین را تا سطح قابل قبولی برآورده سازد، تنظیم کرد و آن را بهبود داد.

علاوه‌براین، بررسی‌های تجربی که انجام می‌دهیم باید آستانه‌هایی را برای ما مشخص کنند که مایل هستیم مقادیر قابلیت استفاده‌ی مجدد از مؤلفه‌ها به آن‌ها برسند. این آستانه‌ها باید منعکس کننده‌ی میزان تلاشی باشند که یک توسعه دهنده مایل به سرمایه گذاری آن است (شبیه به ایده‌ی سنجه‌ی میزان تلاش از هالستید [۱۵]). با این حال، این یک ایده‌ی چالش برانگیز است چراکه ویژگی‌های انتزاعی و فنی از مؤلفه‌ی نرم افزاری باید به ارزش‌های زمانی، هزینه‌ای و یا میزان تلاش ترجمه شوند (به عنوان مثال، «برای استفاده‌ی مجدد از یک مؤلفه در یک زمینه خاص چقدر تلاش نیاز است اگر پیچیدگی آن برابر با X باشد؟»). بدیهی است که در این زمینه تحقیقات بیشتری مورد نیاز است.

مسئله‌ی دیگری که وجود دارد در مورد طبیعت جعبه‌ی سفید و جعبه‌ی سیاه از مؤلفه‌ها است. اگر در مجموعه‌ی مورد نظر ما ترکیبی از چنین مؤلفه‌هایی وجود داشته باشند در این‌صورت مقایسه‌ی قابلیت استفاده‌ی مجدد از آن‌ها ممکن است دشوار شود چراکه در مورد مؤلفه‌های جعبه‌ی سفید محاسبه‌ی سنجه‌ها براساس کل مؤلفه صورت می‌گیرد در حالیکه در مؤلفه‌های جعبه‌ی سیاه محاسبه‌ی سنجه‌ها تنها براساس قسمت‌هایی از مؤلفه که قابل دیدن هستند، صورت می‌پذیرد. بنابراین برای مقایسه‌ی قابلیت استفاده‌ی مجدد از مؤلفه‌های جعبه‌ی سفید و جعبه‌ی سیاه باید تحقیقات بیشتری صورت بگیرد.

۷-مراجع:

[۱] Kakarontzas, G., Constantinou, E., Ampatzoglou, A., & Stamelos, I. (2013). "Layer assessment of object-oriented software: A metric facilitating white-box reuse". Journal of Systems and Software, 86(2), 349-۳۶۶.

[۲] Gill, N. S. (2006). "Importance of software component characterization for better software reusability". ACM SIGSOFT Software Engineering Notes, 31(1), 1-3.

[۳] Mistrík, I., Bahsoon, R., Eeles, P., Roshandel, R., & Stal, M. (Eds.). (2014). "Relating System Quality and Software Architecture". Morgan Kaufmann.

[۴] Kahtan, H., Bakar, N. A., & Nordin, R. (2012, October). "Reviewing the challenges of security features in component based software development models". In Proceedings of the IEEE Symposium on E-Learning, E-Management and E-Services (pp. 1-6).

[۵] Washizaki, H., Yamamoto, H., & Fukazawa, Y. (2003, September). "A metrics suite for measuring reusability of software components". In Software Metrics Symposium, 2003. Proceedings. Ninth International (pp. 211-223). IEEE.

[۶] Singh, Y., Bhatia, P. K., & Sangwan, O. (2011). "Software reusability assessment using soft computing techniques". ACM SIGSOFT Software Engineering Notes, 36(1), 1-7.

[۷] Lingyun, F., Guang, S., & Jianli, C. (2010, July). "An approach for component-based software development". In Information Technology and Applications (IFITA), 2010 International Forum on (Vol. ۱, □□. ۲۲-۲۵). □□□□.

[۸] Hu, K., Guo, Z., Jiang, Y., Feng, Y., & Shen, F. (2012, October). "Component-based development framework for ocean information system". In Oceans, 2012 (pp. 1-7). IEEE.

[۹] Zhang, H. Y., Zhang, L., Urtado, C., Vauttier, S., & Huchard, M. (2012, September). "A three-level component model in component based software development". In ACM SIGPLAN Notices (Vol. 48, No. ۳, □□. ۷۰-۷۹). □□□.

[۱۰] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). "Design patterns: elements of reusable object-oriented software". Pearson Education.

[۱۱] Pressman, R. S. (2005). "Software engineering: a practitioner's approach". Palgrave Macmillan.

[۱۲] Singh, S., Thapa, M., Singh, S., & Singh, G. (2010). Software Engineering-Survey of Reusability Based on Software Component.

[۱۳] Sommerville, I. (2010). "Software Engineering". 9 ed. Addison-Wesley.

[۱۴] Caldiera, G., & Basili, V. R. (1991). "Identifying and qualifying reusable software components". Computer, 24(2), 61-70.

[۱۵] Halstead, M. H. (1977). "Elements of software science". Amsterdam, Elsvier North-Holland.

[۱۶] McCabe, T. J. (1976). "A complexity measure". Software Engineering, IEEE Transactions on, (4), 308-۳۲۰.

[۱۷] Chidamber, S. R., & Kemerer, C. F. (1994). "A metrics suite for object oriented design". Software Engineering, IEEE Transactions on, 20(6), 476-493.

[۱۸] Lee, Y., & Chang, K. H. (2000, April). "Reusability and maintainability metrics for object-oriented software". In Proceedings of the 38th annual on Southeast regional conference (pp. 88-94). ACM.

[۱۹] Cho, E. S., Kim, M. S., & Kim, S. D. (2001, December). "Component metrics to measure component quality". In Software Engineering Conference, 2001. APSEC 2001. Eighth Asia-Pacific (pp. 419-426). IEEE.

[۲۰] Gui, G., & Scott, P. D. (2008, November). "New coupling and cohesion metrics for evaluation of software component reusability". In Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for (pp. 1181-1186). IEEE.

[۲۱] Poulin, J. S. (2006). "The business case for software reuse: Reuse metrics, economic models, organizational issues, and case studies". In Reuse of Off-the-Shelf Components (pp. 439-439). Springer Berlin Heidelberg.

[۲۲] Boxall, M. A., & Araban, S. (2004). "Interface metrics for reusability analysis of components". In Software Engineering Conference, 2004. Proceedings. 2004 Australian (pp. 40-51). IEEE.

[۲۳] Etzkorn, L. H., Hughes, W. E., & Davis, C. G. (2001). "Automated reusability quality analysis of OO legacy software". Information and Software Technology, 43(5), 295-308.

[۲۴] Selby, R. W. (2005). "Enabling reuse-based software development of large-scale systems". Software Engineering, IEEE Transactions on, 31(6), 495-510.

[۲۵] Sandhu, P. S., & Singh, H. (2006). "Automatic reusability appraisal of software components using neuro-fuzzy approach". International Journal Of Information Technology, 3(3), 209-214.

[۲۶] Sandhu, P. S., Blecharz, P., & Singh, H. (2007). "A Taguchi approach to investigate impact of factors for reusability of software components". Transactions on Engineering, Computing and Technology, 19, 135-۱۴۰.

[۲۷] Sandhu, P. S., Kaur, H., & Singh, A. (2009). "Modeling of reusability of object oriented software system". World Academy of Science, Engineering and Technology, 56(32).

[۲۸] Manhas, S., Vashisht, R., Sandhu, P. S., & Neeru, N. (2010). "Reusability evaluation model for procedure based software systems". International Journal of Computer and Electrical Engineering, 2(6), 1107-1110.

[۲۹] Mahmood, A. K., & Oxley, A. (2011). "Reusability Assessment of Open Source Components for Software Product Lines". International Journal of New Computer Architectures and their Applications (IJNCAA), ۱(۳), ۵۱۹-۵۳۳.

[۳۰] Kumar, A. (2012). "Measuring Software reusability using SVM based classifier approach". International Journal of Information Technology and Knowledge Management, 5(1), 205-209.

[۳۱] Hristov, D., Hummel, O., Huq, M., & Janjic, W. (2012, November). "Structuring Software Reusability Metrics for Component-Based Software Development". In ICSEA 2012, The Seventh International Conference on Software Engineering Advances (pp. 421-429).

[۳۲] Barnard, J. (1998). "A new reusability metric for object-oriented software". Software Quality Journal, ۷(۱), ۳۵-۵۰.

[۳۳] Yingmei, L., Jingbo, S., & Weining, X. (2012). "On Reusability Metric Model for Software Component". In Software Engineering and Knowledge Engineering: Theory and Practice (pp. 865-870). Springer Berlin Heidelberg.

[۳۴] Khoshkbarforoushha, A., Jamshidi, P., & Shams, F. (2010, May). "A metric for composite service reusability analysis". In Proceedings of the 2010 ICSE Workshop on Emerging Trends in Software Metrics (pp. 67-74). ACM.

[۳۵] Sharma, A., Kumar, R., & Grover, P. S. (2007). "A critical survey of reusability aspects for component-based systems". World academy of science, Engineering and Technology, 19, 411-415.

[۳۶] Mahmood, A. K., & Oxley, A. (2011). "A Review of Software Component Reusability Assessment Approaches". Research Journal of Information Technology, (3), 1-10.

[۳۷] Ismail, S., Wan-Kadir, W. M., Saman, Y. M., & Mohd-Hashim, S. Z. (2008, August). "A review on the component evaluation approaches to support software reuse". In Information Technology, 2008. ITSim ۲۰۰۸. □□□□□□□□□□□□□□□ □□□□□□□□□□ □□ (□□□. ۴, □□. ۱-۶). □□□□.

[۳۸] Parmeza, D., & Fifo, M. (2013). "A Survey of Methods for Measuring and Enhancing Component Reusability". IDT Mini-conference on Interesting Results in Computer Science and Engineering.

[۳۹] Sandhu, P. S., Kakkar, P., & Sharma, S. (2010, September). "A survey on Software Reusability". In Mechanical and Electrical Technology (ICMET), 2010 2nd International Conference on (pp. 769-773). IEEE.

[۴۰] Subedhaa, V., & Sridharb, S. (2012). "A Systematic Review of Reusability Assessment Model and Related Approach for Reusable Component Mining". Journal of Computer Applications (JCA), 5(2)

[۴۱] Sadana, N., Dhaiya, S., & Ahuja, M. S. "An empirical review on factors affecting reusability of programs in software engineering".

[۴۲] Thakral, S., & Sagar, S. (2014). "Reusability in Component Based Software Development-A Review". World Applied Sciences Journal, 31(12).

[۴۳] Kitchenham, B. (2004). "Procedures for performing systematic reviews". Keele, UK, Keele University, ۳۳(۲۰۰۴), ۱-۲۶.

[۴۴] Keele, S. (2007). "Guidelines for performing systematic literature reviews in software engineering". (pp. ۱-۵۷), Technical report, EBSE Technical report EBSE -2007-01.

[۴۵] Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). "Lessons from applying the systematic literature review process within the software engineering domain". Journal of systems and software, 80(4), 571-583.

[۴۶] Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). "Systematic literature reviews in software engineering–a systematic literature review". Information and software technology, 51(1), 7-15.

[۴۷] Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). "Systematic literature reviews in software engineering–a tertiary study". Information and Software Technology, 52(8), 792-805.

[۴۸] Da Silva, F. Q., Santos, A. L., Soares, S., França, A. C. C., Monteiro, C. V., & Maciel, F. F. (2011). "Six years of systematic literature reviews in software engineering: An updated tertiary study". Information and Software Technology, 53(9), 899-913.

پیوست:

1. Component Based Software Development
2. Reusable Components
3. Reusability
4. Mesurement
24. External Class Complexity
25. Class Coupling Complexity
26. Class Cohesion Complexity
27. Profile Complexity
47. Self-completed
48. Component's Interface
49. Interface Size
50. Identifier Length

5. Systematic
6. Metrics
7. Models
8. Assessment
9. Reuse Costs
10. Functional Usefulness
11. Quality of Components
12. Halstead’s Volume
13. Cyclomatic Complexity
14. Regularity
15. Reuse Frequency
16. Distinct Operators
17. Distinct Operands
18. Program Vocabulary
19. Program Length
20. Calculated Program Length
21. Volume
22. Difficulty
23. Internal Class Complexity
28. Commonality Functions
29. Component Variability
30. Attribute customization
31. Behavior customization
32.Workflow customization
33. Cohesion
34. Coupling
35. Transitive Relationships
36. Existence of Meta-Information
37. Rate of Component Observability
38. Rate of Component Customizability
39. Self-Completeness
40. Component’s Parameter
41. Operational Behaviors
42. Read Methods
43. Properties
44. Faced Class
45. Write Methods
46. Business Methods
51. Argument Count
52. Procedure Declarations
53. Distinct Argument Count and Ratio
54. Argument Repetition Scale
55. Mean String Commonality
56. Functionality
57. Reliability
58. Utilizability
59. Portability
60. Tuned Weighted Methods per Class
61. Lack of Tuned Degree of Inheritance
62. Lack of Tuned Number of Children
63. Lack of Coupling between Objects
64. Lack of Cohesion in Methods
65. Goal-Question-Metric
66. Adjusted number of message sends
67. Friend functions
68. External variable accesses
69. Public functions