نگاه اجمالی به روش های تحلیل و معماری نرم افزار

دکتر ناظمی[1]، مهندس ایزدی، مهندس عمادی، مهندس حیدری، مهندس قزلباش[2]

واژگان کلیدی:
مهندسی نرم افزار، معماری نرم افزار، متدولوژی های نرم افزاری

مقدمه

با توجه به رشد روز افزون سازمان ها و گسترده شدن فرآیندها و عملیات شان از یک سو و پیشرفت روز افزون علوم مرتبط با حوزه فناوری اطلاعات و ارایه روش های نوین در این حوزه به خصوص در زمینه مهندسی نرم افزار، باعث شده است که معماری های مختلفی جهت پیش برد مطلوب کار در زمینه تحلیل و معماری نرم افزار، طراحی و ارایه شود که انتخاب صحیح یکی از این معماری ها در اجرای پروژه های بومی از اهمیت خاصی برخوردار است. در شماره های قبل به بحث پیرامون معرفی برخی از انواع معماری های نرم افزار پرداختیم و در شماره قبل روش های تحلیل معماری نرم افزار **را مورد بررسی قرار دادیم. در این مقاله قصد داریم تا نگاهی اجمالی به روش های مختلف تحلیل مبتنی بر مهندسی نرم افزار را داشته و نقش این روش ها را در توسعه سیستم های اطلاعاتی مورد بررسی قرار دهیم.**

۱- استاد دانشگاه شهید بهشتی
۲- دانشجویان کارشناسی ارشد نرم افزار دانشگاه قزوین

نگاه اجمالی به روش‌های تحلیل

۱- Scenario-Based Architecture Analysis Method (SAAM)

روش تحلیل و طراحی SAAM در سال ۱۹۹۳ پدیدار شد، که مترادف با تمایل برای فهم بهتر مفاهیم کلی معماری بود. بنابراین، در گام‌های اولیه توسعه یک سیستم، تصحیح اشتباهاتی که به‌وسیله تحلیل تشخیص داده می‌شود، بدون صرف هزینه‌های بسیار زیاد امکان پذیر است. فعالیت‌های اصلی این متد در مقاله‌ای جهت ارزیابی واسط‌های معماری مختلف نسبت به تغییرپذیری شان ارایه می‌شود.

اهداف خاص: یک هدف SAAM بررسی صحت و صقم فرضیات و قواعد پایه‌ای معماری در برابر اسنادی که خصوصیات مطلوب یک برنامه کاربردی را توصیف می‌کنند، می‌باشد. به‌علاوه، تحلیل کمک می‌کند تا بتوانیم ریسک‌های ذاتی معماری را ارزیابی کنیم. SAAM ما را در نظارت معماری با تمرکز بر روی نقاط بحرانی راهنمایی می‌کند، از جمله این نقاط بحرانی تضاد بین نیازمندی‌ها یا مشخصه طرح غیرکامل از دید یک stakeholder می‌باشد. توانایی SAAM برای ارزیابی مناسب بودن معماری در خصوص خصوصیات مطلوب یک سیستم ویژه می‌تواند همچنین برای مقایسه معماری‌های مختلف استفاده شود.

تکنیک ارزیابی: سناریوها پایه‌ای را برای روشن کردن خصوصیات SA ارایه می‌کنند. آن‌ها انواع فعالیت‌هایی که سیستم باید پشتیبانی کند و انواع تغییرات پیش بینی شده‌ای که برای سیستم رخ خواهد داد را نشان می‌دهند. در مدت زمانی که تحلیل انجام می‌شود، مشخص می‌شود که آیا یک سناریو، اصلاحاتی در جهت معماری نیاز دارد. سناریوهایی که نیاز به اصلاحات ندارند direct و آن‌هایی که نیاز به اصلاحات دارند indirect نامیده می‌شوند.

خصوصیات کیفیتی[1]**:** خصوصیت بنیادی این متد، ساخت خصوصیت کیفیتی در فرم سناریو می‌باشد.

Stakeholderهای درگیر: SAAM علاقمندی‌های مختلف گروه‌های stakeholder را هماهنگ می‌کند، بنابراین یک فهم عمومی از SA برای تصمیمات بعدی ایجاد می‌کند.

- **توصیف SA:**

این متد برای ویرایش نهایی SA اعمال می‌شود و این در حالیست که طراحی جزئی انجام شده است. توصیف SA باید به فرمی باشد که به آسانی بوسیله همه stakeholderها قابل فهم باشد. Functionality, Structure و Allocation سه دیدگاهی هستند که برای توصیف SAها تعریف می‌شوند. Functionality کاری است که سیستم انجام می‌دهد. یک لغت نامه کوچک و ساده برای توصیف Structureها استفاده می‌شود که از یک سطح عمومی برای فهم و مقایسه معماری‌های مختلف برخوردار می‌باشد. SA هم باید از جنبه ایستا (محاسبات سیستم، داده‌های اجزا، ارتباطات داده‌ای و کنترلی) نمایش داده شود و هم از جنبه پویا که بیان‌گر چگونگی رفتار سیستم در هر زمان می‌باشد، نمایش داده شود. در ساخت یک تابع allocation مشخص می‌کند که چگونه دامنه functionality در ساختار نرم‌افزار فهمیده می‌شود. اجزا می‌توانند یا به عنوان ماژل توصیف شوند که نظر Parnas می‌باشد و یا به عنوان فرآیندهای پشت سر هم که با همکاری هم کار می‌کنند، توصیف شوند.

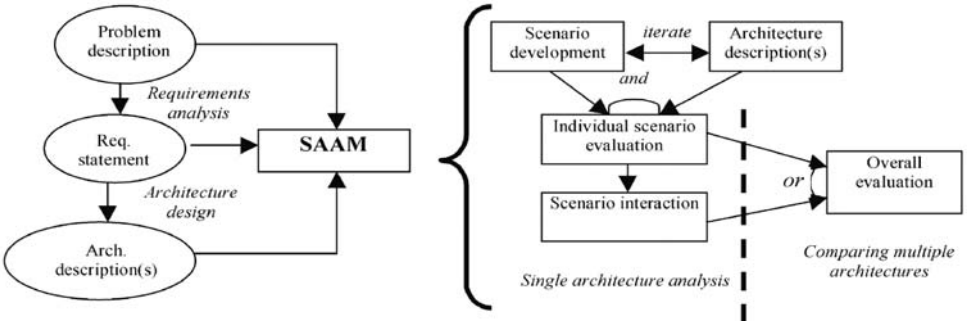
- **فعالیت‌های متد:**

ورودی‌های اصلی SAAM توصیف مساله، نیازمندی‌ها و توصیف معماری می‌باشند. شکل۱ ورودی‌های مورد نیاز برای فعالیت‌های SAAM را نمایش می‌دهد که یا برای یک معماری منفرد و یا برای مقایسه چندین معماری مورد نیاز هستند.

در مورد تحلیل یک SA منفرد، فعالیت‌ها عبارتند از توسعه سناریو، توصیف SA، ارزیابی سناریوی شخصی و فعل و انفعال سناریو. در توسعه سناریو، SAAM نیاز به حضور همه stakeholderها دارد تا امکان پذیری سناریوها را همان‌طوری که در بالا توصیف شد، مشخص کنند. SAAM یک مجموعه از سناریوها را برای کامل شدن مورد بررسی قرار می‌دهد، زمانی که افزایش یک سناریوی جدید برای مدت طولانی طراحی را مختل نمی‌کند. توصیه می‌شود دومین فعالیت یعنی توصیف SA، بطور موازی با اولین فعالیت در یک مد تعاملی انجام شود. ویرایش نهایی توصیف SA، به اضافه سناریوها به عنوان ورودی برای فعالیت‌های بعدی متد ایفای نقش می‌کنند.

SAAM یک سناریو را با مورد توجه قرار دادن آن دسته از عناصر معماری که سناریو بر آن‌ها اثر گذار است، ارزیابی می‌کند. جدول۱ مثالی از ارزیابی سناریو برای یک معماری می‌باشد که این معماری شامل کامپوننت‌هایی است که A، B, C, D و E نامیده می‌شوند. در تحلیل یک معماری منفرد هدف مشخص کردن مواردی است که سناریو با آن‌ها کار می‌کند. هزینه تغییرات مربوط به هر سناریوی indirect بوسیله لیست کردن کامپوننت‌ها و اتصال‌های موثر و سپس شمردن تعداد تغییرات، تخمین زده می‌شود. اگر تحلیل به‌منظور انتخاب از میان چندین معماری انجام می‌شود، نتیجه

1-Conflict



شکل۱: ورودی‌ها و فعالیت‌های SAAM

به‌دست آمده از این تحلیل می‌تواند در فعالیت نهایی SAAM مقایسه شود. در پایان سناریوها و فعل و انفعالات سناریوها برحسب اهمیت نسبی شان وزن دهی می‌شوند. سپس از این وزن دهی برای مشخص کردن یک ارزیابی کلی از معماری‌های انتخاب شده استفاده می‌شود.

Stakeholder	*Scenario*	*Scenario Description*	*Direct/ Indirect*	*Architecture Changes*
User	U1	....	Indirect	Modifications to A and B components
	U2	....	Direct	-
Maintainer	M1	...	Indirect	Modifications to A, B, C, D and E components
	M2	....	...	...
Administrator	A1	...	...	...
	A2	...	...	...

جدول۱: ارزیابی سناریو

- **تفسیر نتایج[1]:**

حجم زیاد فعل و انفعالات نامرتبط سناریوها می‌تواند جدایی ضعیف functionality را مشخص کند. میزان فعل وانفعال سناریو مربوط به متریک‌هایی مانند پیچیدگی ساختاری، اتصال و انسجام می‌شود و بنابراین به شکل نقایصی در محصول نهایی نمایان می‌شود. SAAM نمی‌تواند اندازه‌های خیلی دقیق یا متریک‌های شایسته‌ای را ارایه دهد. استفاده مجدد از دانش موجود: SAAM این موضوع را مورد بررسی قرار نمی‌دهد.

- **اعتبارسنجی متد[2]:**

SAAM یک متد کامل می‌باشد که در مطالعات موردی مختلف صحت آن مورد بررسی قرار گرفته است.

۲- SAAM پایه گذاری شده بر روی سناریوهای پیچیده[3] (SAAMCS)

بحث SAAMCS در مورد پیچیدگی سناریوها می‌باشد که فاکتور بسیار مهمی برای تشخیص ریسک می‌باشد. سهم SAAMCS برای توسعه SAAM از طرفی مربوط به روش جستجوی سناریوها و از طرف دیگر به بحث ارزیابی مربوط می‌شود.

- **هدف خاص:**

تشخیص ریسک هدف مهم SAAMCS را نمایش می‌دهد.

- **خصوصیات کیفیتی:**

انعطاف پذیری، خصوصیت کیفیتی خصوصیت تحلیل شده به‌وسیله SAAMCS را ارایه می‌کند.

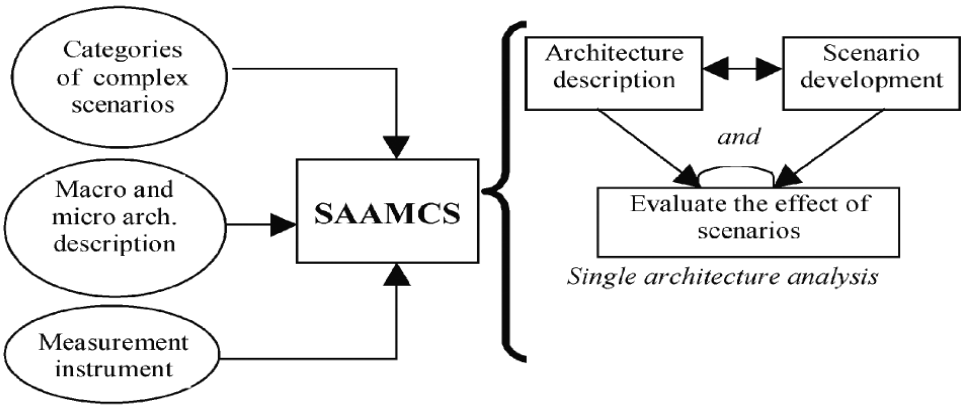
- **Stakeholderهای درگیر:**

متد به stakeholderهای درگیر بها می‌دهد و نقش مهم آغازگر سناریو را مشخص می‌کند. آغازگر، واحد سازمانی می‌باشد که بیشترین نفع را از پیاده سازی سناریو دارد.

1 - Result interpretation
2 - Validation
3- SAAM Founded on Complex Scenario

● **توصیف SA:**

SAAMCS برای ویرایش نهایی معماری بکار برده می‌شود، که این ویرایش با جزئیات کافی توصیف می‌شود. در این متد از سیستم‌ها درون یک دامنه ایزوله نمی‌شوند اما در عوض درون محیطی که توسعه داده می‌شود مجتمع می‌شوند. به عنوان یک نتیجه، توصیف SA در دو بخش Macroarchitecture و Microarchitecture تقسیم می‌شود.



شکل۲: ورودی‌ها و فعالیت‌های **SAAMCS**

● **فعالیت‌های متد:**

شکل۲ ورودی‌ها و فعالیت‌های SAAMCS را شرح می‌دهد. در توسعه سناریو، یک دیاگرام دو بعدی (پنج طبقه بندی از سناریوهای پیچیده، چهار منبع تغییرات) که ممکن است برای پوشش دادن سناریوهای پیچیده کمک کند، تعریف می‌شود. منابع تغییرات نیازمندی‌های پایه، نیازمندی‌های کیفیتی، کامپوننت‌های خارجی و محیط فنی هستند. طبقه بندی‌های سناریوهای پیچیده بر اساس سازگاری سیستم با اثرات جانبی و سازگاری محیط با اثراتی که بر سیستم دارد، می‌باشد.

راجع به ارزیابی برخورد سناریو، SCMAAS یک ابزار اندازگیری را معرفی می‌کند و برای بیان اثر سناریوها به‌کار می‌برد. ابزار معرفی شده شامل فاکتورهایی است که بر پیچیدگی سناریوها اثر می‌کنند. سه فاکتور متفاوت معرفی می‌شوند: چهار سطح برخورد سناریو (۱ـ برخورد وجود ندارد، ۲ ـ اثرات یک کامپوننت، ۳ ـ اثرات چندین کامپوننت، ۴ ـ اثرات SA)، تعداد مالکان درگیر در سیستم اطلاعاتی و چهار سطح راجع به وجود ناسازگاری‌های ویرایشی (۱ـ عدم وجود مشکل در ویرایش‌های مختلف، ۲ ـ یک مورد نامطلوب وجود دارد اما منعی ندارد، ۳ ـ ایجاد پیچیدگی‌های مربوط به مدیریت پیکربندی، ۴ ـ ایجاد ناسازگاری‌ها). نتایج در جدولی مشابه جدول ۳ بیان می‌شوند.

● **استفاده مجدد از دانش پایه‌ای موجود:**

SAAMCS این موضوع را مورد بحث قرار نمی‌دهد.

● **اعتبارسنجی متد:**

صحت و سقم SAAMCS برای سیستم‌های اطلاعاتی تجاری بررسی شده‌اند.

Change scenario	*Initiator*	*Macro architecture level*			*Micro architecture level*		
		Impact level	*Multiple owners*	*Versions conflicts*	*Impact level*	*Multiple owners*	*Versions conflicts*
S1	User	1	+/-	4	2	+/-	4
…	…	…	…	…	…	…	…

جدول۲: نتایج و ارزیابی سناریوی **SAAMCS**

۳- Extending SAAM by Integration in the Domain (ESAAMI)

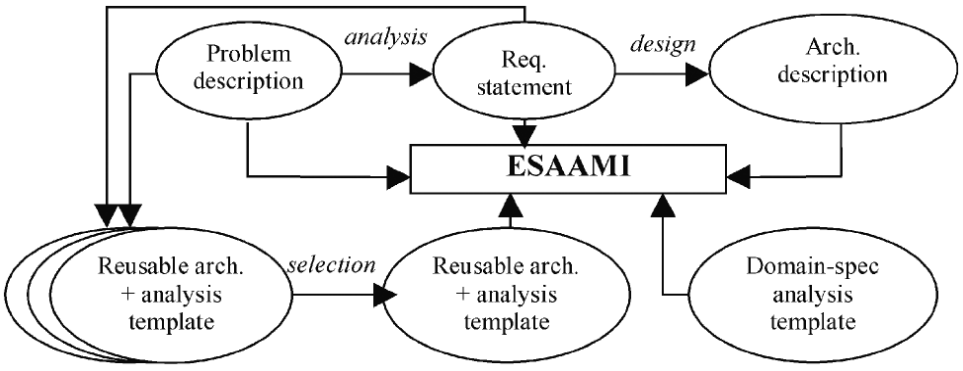
● **هدف خاص:**

وقتی که در فرآیند توسعه معماری از SAAM استفاده شود فقط توصیف مساله، بیان نیازمندی‌ها و توصیف معماری

مورد رسیدگی قرار می‌گیرند.
ESAAMI ترکیبی از مفاهیم تحلیل و استفاده مجدد می‌باشد و با اجتماع SAAM در دامنه خاص و فرآیند توسعه بر مبنای استفاده مجدد به‌دست می‌آید. درجه استفاده مجدد به‌وسیله تمرکز بر روی دامنه اصلاح می‌شود. ESAAMI شبیه SAAM می‌باشد که در تکنیک ارزیابی، خصوصیات کیفیتی، stakeholder درگیر و توصیف SA دارای ملاحظاتی می‌باشد. هرچند، اصلاحاتی نیز در استفاده مجدد از دامنه دانش تعریف شده به‌وسیله SAها و الگوهای تحلیل دیده می‌شود. شکل ۳ ورودی‌های اصلی ESAAMI و روابط بین آن‌ها را شرح می‌دهد. یک SA با قابلیت استفاده مجدد به همراه یک الگوی مناسب برای تحلیل که بر روی خصوصیات مشخصی از معماری تمرکز دارد، بسته بندی می‌شود. همه این بسته‌ها ورودی‌هایی را جهت فرآیند انتخاب معماری باقابلیت استفاده مجدد ارایه می‌کنند. SA انتخاب شده یک نقطه شروع برای طراحی معماری می‌باشد، که برای رویارویی با خصوصیات سیستم جدید تطبیق و تعریف شده است.

- **توصیف SA:**

یک SA با قابلیت استفاده مجدد برای توسعه داده شدن در یک سیستم جدید در اولین گام ESAAMI انتخاب می‌شود. باید اطمینان حاصل شود که SA یک پایه مناسبی را برای سیستم جهت برآورده شدن نیازمندی‌های آن ارایه می‌کند. سه فاکتور در قابلیت استفاده مجدد یک معماری اثر بخش است. نویسنده یک پایه عمومی برای تنوع سیستم‌ها در یک دامنه تعیین می‌کند.



شکل ۳: ورودی‌های **ESAAMI**

- **استفاده مجدد از دانش پایه‌ای موجود:**

ESAAMI بسته‌های الگوهای تحلیلی را پیشنهاد می‌کند که خصوصیات اساسی دامنه را ارایه می‌کنند. یک الگوی تحلیل بر روی یک سطح تجریدی فرموله بندی می‌شود که به‌وسیله افراد عادی سیستم‌های موجود در دامنه و بدون رجوع به عناصر معماری سیستم خاص، تعریف شده است. الگوهای تحلیل محصولات با قابلیت استفاده مجدد را جمع می‌کنند که می‌توانند در گام‌های مختلف متد توسعه داده شوند. این محصولات سناریوهای اولیه، پروتکل‌های ارزیابی، ارزیابی‌های اولیه و اشاره‌ها [1] و وزن‌های معماری می‌باشند. سناریوهای اولیه توصیفات عمومی‌از وضعیت‌های قابل استفاده مجدد یا تعاملات با سیستم هستند. محصولات دیگر در مرحله ارزیابی سناریو استفاده می‌شوند و در پروتکل‌های ارزیابی‌های اولیه در پروژه‌های مختلف تعیین می‌شوند، مثال‌هایی از توصیف این‌که سناریو چطور می‌تواند با به‌کارگیری یک مجموعه از عناصر انتزاعی معماری انجام شود، و اشاره‌ها با هر سناریو همکاری می‌کنند برای نشان دادن این‌که کدام ساختارها سناریوی مناسب را برای بکار بردن می‌خواست بسازد. وزن‌ها که در پروژه‌های قبلی به‌دست آمده‌اند، می‌توانند نتایج تحلیل را قابل مقایسه کنند.

- **فعالیت‌های متد:**

مشابه با SAAM هستند، با این تفاوت که آن‌ها وجود دانش پایه با قابلیت استفاده مجدد را بررسی می‌کنند. نتایج تحلیل جاری بخشی از سیستم تازه ساخته شده می‌باشند.

اعتبارسنجی متد:

متد هنوز در حال بهبود می‌باشد.

1- Hint

۴ـ متد تحلیل معماری نرم‌افزار برای تکامل و قابلیت استفاده مجدد (SAAMER)

- **هدف خاص:**

SAAM از نقطه نظر دو خصوصیت کیفیتی عملی یعنی تکامل و قابلیت استفاده مجدد، در SAAMER توسعه داده می‌شود. SAAMER پیشنهادات خوبی را در این خصوص که چطور یک سیستم می‌تواند هر کدام از اهداف کیفیتی یا سطوح ریسک را پشتیبانی کند، و یا این‌که چطور آن را قابل استفاده مجدد کند ارایه می‌کند.

- **تکنیک ارزیابی:**

سناریوها راه‌اندازهای اصلی ارزیابی فضاهای مختلف SA می‌باشند. آن‌ها عملیات مهمی را که سیستم باید پشتیبانی کند را شرح می‌دهند، یا تشخیص می‌دهد که کجا سیستم ممکن است نیاز به تغییر دادن داشته باشد. سناریوها بر مبنای اهداف stakeholderها و معماری توسعه داده می‌شوند و کاربردهای پایه‌ای سیستم را مورد بررسی قرار می‌دهند. سناریوها و دیدگاه ساختاری در مشخص کردن کامپوننت‌هایی که نیاز به اصلاح شدن دارند موثر می‌باشند، یا برای فعالیت‌های پشتیبانی جلوگیری کننده و سازگار کننده سودمند می‌باشند.

- **خصوصیات کیفیتی:**

تکامل و قابلیت استفاده مجدد مورد رسیدگی قرار داده می‌شوند. تکامل، اهداف کیفیتی جدید (قابلیت پشتیبانی و اصلاح پذیری) را از طریق افراد خبره در دامنه مساله جمع آوری می‌کند.

- **Stakeholderهای درگیر:**

مشابه SAAM می‌باشد. به‌علاوه دو نوع از منابع اطلاعاتی یعنی تغییرات مورد نیاز و تجربیات افراد خبره در دامنه مساله مورد رسیدگی قرار داده می‌شوند.

- **توصیف SA:**

SAAMER, viewهای مهم معماری را که در زیر اشاره شده است را مورد رسیدگی قرار می‌دهد. static, map, dynamic و resource. Static view, SAAM را برای طبقه بندی کردن و عمومیت دادن کامپوننت‌ها و توابع سیستم و روابط بین آن‌ها مجتمع می‌کند و توسعه می‌دهد. این توسعه‌ها تخمین هزینه یا تلاش مورد نیاز برای ایجاد تغییرات را آسان می‌کنند. Dynamic view مناسب ارزیابی جنبه رفتاری برای بررسی صحت و سقم کنترل و ارتباطات به‌کار رفته، به روش مورد انتظار می‌باشد. نگاشت بین کامپوننت‌ها و توابع می‌تواند به مفاهیم انسجام واتصال در یک سیستم وابسته باشد.

- **فعالیت‌های متد:**

SAAMER یک چهارچوب از فعالیت‌ها را ارایه می‌کند که برای فرآیند تحلیل سودمند می‌باشند. این چهارچوب ترکیبی است از چهار فعالیت که عبارتنداز: گردآوری اطلاعات درباره Stakeholderها، SA، کیفیت، سناریوها؛ مدل کردن مصنوعات قابل استفاده؛ تحلیل؛ ارزیابی. دو فعالیت آخر مشابه با SAAM هستند. هرچند، در فاز توسعه سناریوی SAAMER، یک جواب سودمند به سوال، در خصوص متوقف شدن ایجاد سناریوها داده می‌شود. در این‌جا دو تکنیک به‌کار برده می‌شود. اولاً، ایجاد سناریو بدقت به انواع مختلف هدف‌ها گره زده می‌شود: Stakeholder، معماری وکیفیت. بر مبنای هدف‌ها و دانش افراد خبره موجود در دامنه، سناریوها تعیین شده و دسته بندی می‌شوند تا اطمینان حاصل شود که هر هدفی بخوبی پوشش داده شده است. دومین تکنیک برای اعتبارسنجی تعادل سناریوها نسبت به هدف (QFD) به‌کار برده می‌شود. از اهداف Stakeholder و معماری برای خصوصیات کیفیتی، یک سری آبشاری از متریک‌ها برای نشان دادن استحکام روابط ایجاد می‌شود. در نهایت، خصوصیات کیفیتی به سناریوها برای مشخص شدن پوشش هر کدام ترجمه می‌شوند. سپس یک فاکتور عدم تعادل برای هر خصوصیت کیفیتی به‌وسیله تقسیم پوشش با اولویت کیفیت محاسبه می‌شود. اگر فاکتور کمتر از ۱ است، بیشتر سناریوها باید برای دستیابی به خصوصیت مطابق Stakeholder, SA، و اهمیت کیفیت توسعه داده شوند.

تفسیر نتایج:

تحلیل فعل و انفعال سناریو یک گام اصلی است برای بررسی شدن آن چیزی است که نتیجه تحلیل را ارایه می‌کند. یک درجه بالا یی از تعامل ممکن است بیان‌گر این باشد که کامپوننت به‌طور ناچیز ایزوله می‌شود. هنوز، ممکن است یک SA view نشان دهد که این فقط طبیعت الگوی عملی می‌باشد.

استفاده مجدد از دانش پایه‌ای:
SAAMER این موضوع را بررسی نمی‌کند.
این متد برای یک سیستم نرم‌افزاری ارتباطاتی به‌کار برده شده است.

۵- The Architecture Trade-off Analysis Method (ATAM)

- **ATAM خارج از کار روی تحلیل معماری مشخصه‌های کیفیتی شخصی رشد کرده است:**

SAAM برای تحلیل Modifiability, Performance, Availability و Security می‌باشد. ATAM یک مدل مارپیچی از طراحی‌ها در ۱۹۹۸ و در May ۱۹۹۹ یک مدل مارپیچی از تحلیل و طراحی مورد رسیدگی قرار داده شد، که توسعه و پیشرفت اخیر آن را نشان می‌دهد.

- **اهداف ویژه:**

هدف ATAM ارایه یک روش اصولی جهت فهم توانایی SA در مقایسه چندین مشخصه کیفیتی می‌باشد. ATAM ضرورت یک Trade-Off میان چندین مشخصه کیفیتی نرم‌افزار را در زمان مشخص کردن SA و قبل از توسعه آن تعیین می‌کند.

- **مشخصه‌های کیفیتی:**

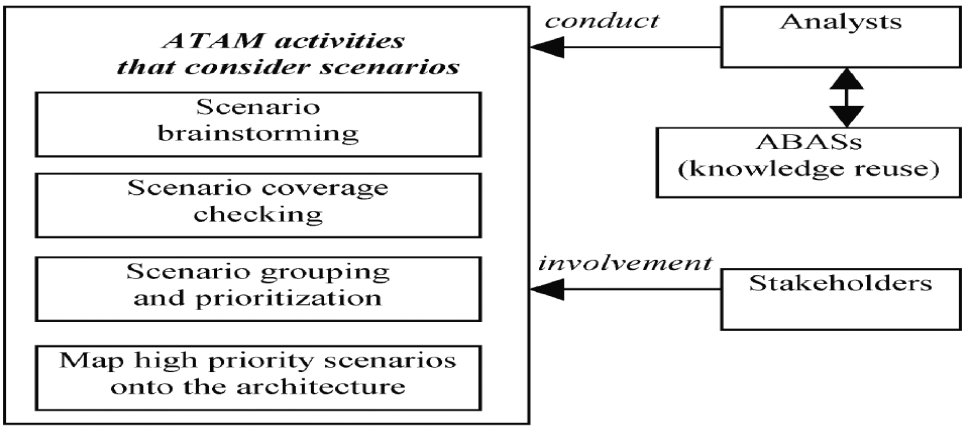
مقایسه چندین مشخصه کیفیتی به‌وسیله ATAM مورد تحلیل قرار داده می‌شود. برای شروع Modifiability, Security, Performance و Availability مورد رسیدگی قرار داده شده است.

- **افراد درگیر:**

ATAM به همه افراد درگیر، در فعالیت‌های مربوط به جمع آوری سناریوها و نیازمندی‌ها نیاز دارد. یک طراح SA نیز می‌تواند شرکت داشته باشد.

- **توصیف SA:**

فضای معماری به‌وسیله Legacy Systems توانایی در در ردو بدل کردن اطلاعات و خطاهای پروژه‌های قبلی محدود می‌شود. SA بر مبنای پنج ساختار پایه‌ای توصیف می‌شود، که از "Kruchten's "4+1view مشتق می‌شوند (جنبه منطقی آن توابع و کدها تقسیم می‌شوند). ساختارها باضافه نگاشت اختصاص یافته بین آن‌ها می‌تواند یک معماری را بطور کامل شرح دهد. همچنین ATAM به چندین View مختلف نیاز دارد: Dynamic View، نشان دهنده چگونگی برقراری ارتباط میان سیستم‌ها ست؛ System View، نشان دهنده این است که چطور نرم‌افزار به سخت افزار اختصاص داده می‌شود؛ و Source View، نشان دهنده این است که کامپوننت‌ها و سیستم‌ها چطور از اشیا ساخته می‌شوند. توصیف SA با یک مجموعه از پیغام مربوط به چارت‌های پشت سر هم[1] که نشان دهنده تعاملات و سناریوهای زمان اجرا هستند به‌طور تفسیری شرح داده می‌شود. ATAM در مدت طراحی SA یا روی ویرایش نهایی SA به‌وسیله یک تیم تحلیل خارجی به‌کار برده می‌شود.



شکل ۴: فعالیت‌های ATAM که سناریوها را بررسی می‌کنند.

1- A set of message sequences charts

● **تکنیک‌های ارزیابی موجود:**

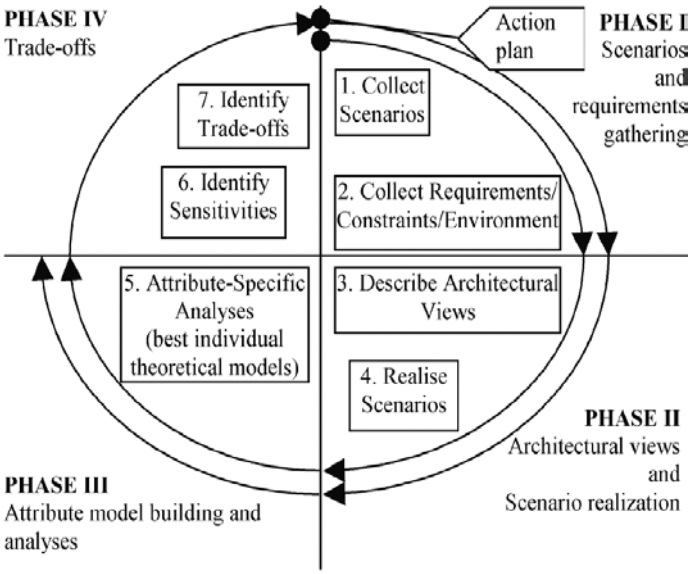
ATAM به عنوان یک چهار چوب برای تکنیک‌های ارزیابی مختلف بسته به خصوصیات کیفیتی مورد بررسی قرار داده شود. ATAM بهترین مدل تئوری شخصی از هر صفت مشخصه بررسی شده را در یک روش موثر و عملی مجتمع می‌کند.

تکنیک دیگر ارزیابی سناریو می‌باشد. سه نوع سناریو سیستم را از جنبه‌های مختلف معماری کاوش می‌کند. این سه نوع عبارتند از: Use Cases، که درگیر کاربردهای نوعی سیستم هستند و برای استخراج اطلاعات استفاده می‌شوند، Growth Scenarios، که تغییرات و پیش بینی کردن را پشتیبانی می‌کند؛ و Exploratory Scenarios، تغییرات زیادی که انتظار می‌رود باعث استرس سیستم شود را پوشش می‌دهد. در این متد سناریو نقش سه گانه دارد. این تکنیک جهت پوشش دادن نیازمندی‌های مبهم و دارای کیفیت نامشخص کمک می‌کند. همچنین سناریوها ارتباطات بین افراد ذی نفع را آسان می‌کنند و این امر به این دلیل است که افراد را مجبور می‌کنند که به یک درک مشترک از نیازمندی‌ها برسند.

ATAM همچنین Qualitative Analysis Heuristics را مورد بررسی قرار می‌دهد، که از یک Architecture Style (ABAS) Attribute-based مشتق می‌شود.

● **فعالیت‌های متد:**

این متد به چهار فضا یا فاز فعالیتی اصلی تقسیم می‌شود. شکل ۴ جزییات فعالیت‌هایی که شامل سناریوها می‌شوند را نشان می‌دهد و شکل ۵ گام‌های مربوط به هر فاز را توصیف می‌کند و تکرارهای ممکن برای طراحی SA و اصلاح تحلیل را توصیف می‌کند.



شکل ۵: فازهای ATAM

● **استفاده مجدد از یک دامنه دانش پایه‌ای:**

در قسمت ABASها پشتیبانی می‌شود.

این متد روی چندین سیستم نرم‌افزاری بکار برده شده است اما هنوز در دست تحقیق است.

۶- Scenario_Based Architecture Reengineering (SBAR)

این متد فقط در طراحی استفاده نمی‌شود بلکه در ارزیابی کیفیت معماری نرم‌افزاری که به‌صورت جزیی بیان‌گر سیستم است بر مبنای سناریو استفاده می‌شود.

● **هدف ویژه:**

SBAR پتانسیل معماری طراحی شده را برای رسیدن به نیازمندی‌های کیفیتی نرم‌افزار تخمین می‌زند.

● **تکنیک ارزیابی استفاده‌شده توسط متد:**

چهار تکنیک مختلف برای تعیین مشخصه‌های کیفیتی مشخص می‌شود: سناریوها، شبیه سازی، مدل سازی ریاضی و استدلال بر مبنای تجربه. برای هر مشخصه کیفیتی تکنیک مناسب انتخاب می‌شود. توصیه می‌شود که سناریوها برای مشخصه‌های کیفیتی که مربوط به توسعه هستند استفاده شود، مانند Maintainability، Reusabilityکه در مقاله با مثال نشان داده شده‌اند. کارایی معماری به‌وسیله سناریوی مربوط به هر مشخصه کیفیتی تعریف می‌شود که در تحلیل مشخص شده است. شبیه سازی، روش‌های برپایه سناریو راکامل می‌کند، برای ارزیابی مشخصه‌های عملی کیفیتی نرم‌افزار مثل کارایی و تحمل پذیری خطا مفید می‌باشد. مدل‌های ریاضی یک ارزیابی ایستایی از مدل‌های طراحی معماری ارایه می‌کنند و یک آلترناتیو برای شبیه سازی هستند. از این‌رو هر دو روش اصولا مناسب برای تعیین مشخصه‌های کیفیتی عملی نرم‌افزار می‌باشند. برای ارزیابی مشخصه‌های کیفیتی عملی نرم‌افزار مدل‌های ریاضی موجود به‌وسیله انجمن‌های تحقیقاتی مختلف توسعه داده می‌شوند تا بتوانند برای High Performance Computing، Reliability و سیستم‌های Real-Time استفاده شوند. استدلال برپایه تجربه بر روی تجربه و استدلال منطقی روی آن تجربه بنا نهاده شده است. این تکنیک متفاوت از دیگر تکنیک‌ها می‌باشد به دلیل این‌که صراحت کمتری دارد و بیشتر روی فاکتورهای ذهنی مانند درک مستقیم و تجربه پایه گذاری شده است و از دانش ضمنی مردم استفاده می‌کند.

● **مشخصه‌های کیفیتی:**

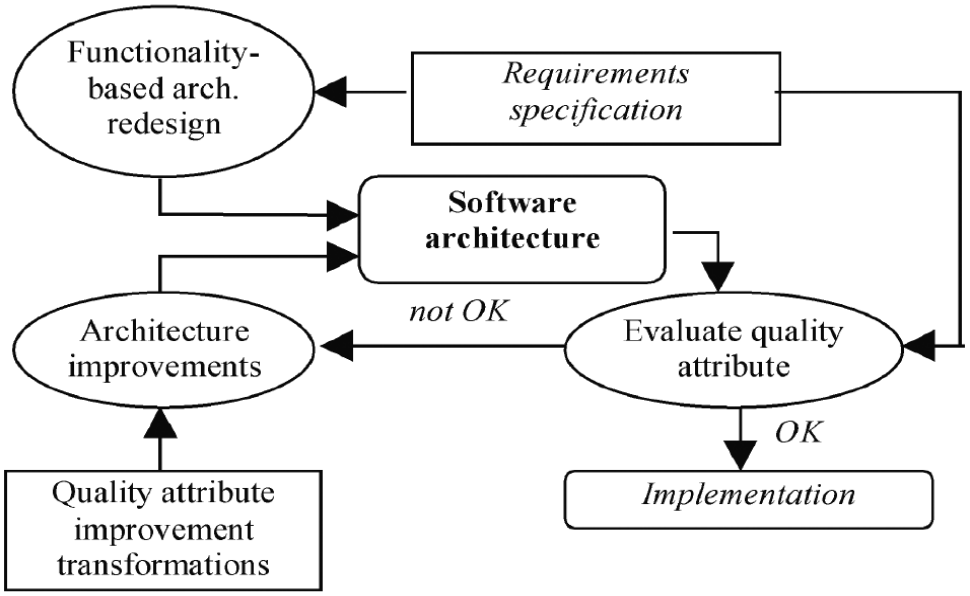
SBAR بر روی چندین مشخصه کیفیتی نرم‌افزار متمرکز می‌شود. تعدادی از انجمن‌های تحقیقاتی که در زمینه مشخصه کیفیتی تحقیق می‌کنند روش‌هایی را متعلق به خودشان برای توسعه سیستم‌های Real-Time، Performance High و Reusable پیشنهاد کرده‌اند. همه این روش‌ها روی یک مشخصه کیفیتی معینی متمرکز می‌شوند. SBAR این روش‌ها راکه رضایت بخش نیستند مورد رسیدگی قرار می‌دهد تا یک تعادل بین مشخصه‌های کیفیتی مختلف که در طراحی هر سیستم واقعی مورد نیاز می‌شود، برقرار کند.

● **افراد درگیر:**

SBAR نیازی به درگیر بودن Stakeholderهای زیاد ندارد. در این‌جا فردی که ارزیابی را انجام می‌دهد طراح SA می‌باشد.

● **توصیف SA:**

یک خصوصیت برجسته این متد آن است که برای ارزیابی معماری سیستم موجود، خود سیستم می‌تواند استفاده شود. SBAR یک طرح دارای جزییات از SA را استفاده می‌کند.



شکل۶: فعالیت‌های SBAR. تحلیل و طراحی SA.

● **فعالیت‌های متد:**

فرآیند ارزیابی ترکیبی است از مشخص کردن مجموعه‌ای از سناریوها برای هر مشخصه کیفیتی نرم‌افزار، اجرای دستی سناریوها روی معماری و تفسیر نتایج (شکل ۶). متد می‌تواند به روش کامل[1] یا آماری[2] انجام شود. در روش اول، مجموعه‌ای از سناریوها تعریف و با هم ترکیب می‌شوند، این سناریوها نمونه‌های به‌هم پیوسته‌ای از مشخصه‌های کیفیتی نرم‌افزار را پوشش می‌دهند. اگر همه سناریوها بدون مشکل اجرا شوند، مشخصه‌های کیفیتی معماری بهینه می‌باشد. دومین روش تعریف کردن مجموعه‌ای از سناریوها است که یک نمونه ساده بدون همه موارد را می‌سازد. واضح است که هر دو روش دارای معایبی می‌باشند. یک عیبی که اولین روش دارد این است که عموما تعریف کردن یک مجموعه کامل از سناریوها غیر ممکن است. تعریف مجموعه‌ای از سناریوها که نمایانگر کل مجموعه باشد نقطه ضعف روش دوم است، از این رو تصمیم گیری روی مجموعه سناریوی که نمایانگر کل مجموعه باشد امری مجهول است. نتایج حاصل از هر تحلیل روی معماری و سناریو در داخل نتایج کلی خلاصه می‌شود، برای مثال تعداد سناریوهایی که پذیرفته شده است متفاوت از تعدادی است که پذیرفته نشده است. SBAR راهنمایی هایی را برای اصلاح روی SA ارائه می‌کند. ساختاری مشابه جدول ۳ برای بیان نتایج سازمان‌دهی می‌شود.

Quality attribute	*Scenario*	*Iteration number*			
		0	*1*	...	*n*
QA1	A1	-			+
	A2	-			+
	...	-	-	-	+
QA2	B1	+	+	+	+
	B2	-	-	-	-
	...	-	-	+	+
...	....	...	...	...	...

جدول ۳: سازماندهی نتایج SBAR

● **استفاده مجدد از دانش پایه‌ای:**

SBAR این مورد را مورد بررسی قرار نمی‌دهد.

SBAR به عنوان متدی برای اندازه گیری سیستم نرم‌افزاری شناخته شده است.

۷- Architecture Level Prediction of Software Maintenance (ALPSM)

● **هدف ویژه:**

ALPSM، Maintainability یک سیستم نرم‌افزاری را به‌وسیله بررسی پیچیدگی سناریوها در سطح SA تحلیل می‌کند. مشابه با Software Maintenance Community، ALPSM از سایز تغییرات به عنوان یک عامل پیش‌گویی کننده برای تلاش مورد نیاز جهت سازگار کردن سیستم در داخل یک سناریو استفاده می‌کند.

● **تکنیک ارزیابی استفاده شده:**

ALPSM یک Maintenance Profile را تعریف می‌کند، مشابه مجموعه‌ای از سناریوهای تغییرات که نمایانگر وظایف مربوط به پشتیبانی کامل و سازگار هستند. یک سناریو یک فعالیت یا مجموعه‌ای از فعالیت‌ها را توصیف می‌کند که ممکن است در ارتباط با سیستم رخ دهند. از این رو، یک سناریوی تغییرات یک وظیفه پشتیبانی معینی را شرح می‌دهد.

● **افراد درگیر:**

فقط طراح در فعالیت‌های مربوط به متد درگیر می‌باشد.

● **توصیف SA:**

ALPSM روی ویرایش نهایی SA به‌کار برده می‌شود.

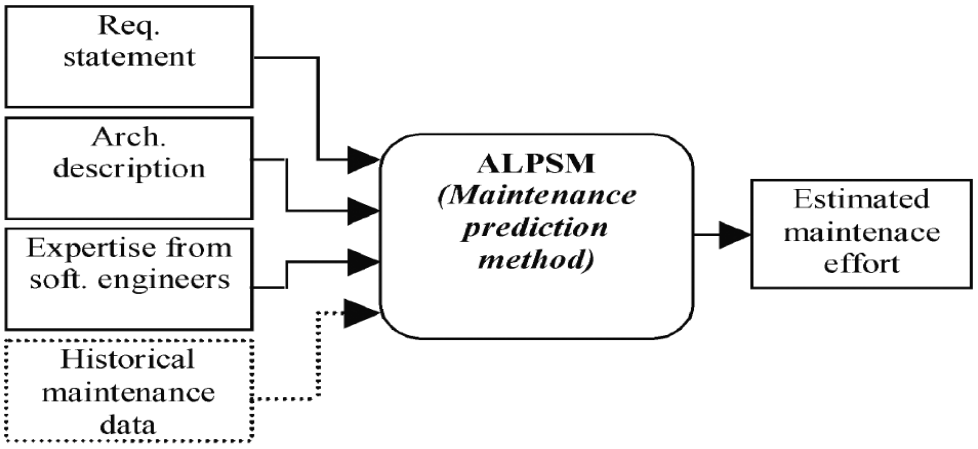
1-Complete
2-Statistical

● **فعالیت‌های متد:**

متد تعدادی ورودی دارد که عبارتند از: بیان نیازمندی‌ها، توصیف معماری، اظهار نظر فنی مهندسان نرم‌افزار و Data Historical Maintenance (شکل ۷). ALPSM ترکیبی از شش گام زیر است:

۱ـ شناسایی طبقه بندی‌های وظایف مربوط به پشتیبانی
۲ـ ترکیب سناریوها
۳ـ نسبت دادن وزن به هر سناریو
۴ـ تخمین اندازه همه عناصر
۵ـ تهیه متن سناریوها
۶ـ محاسبه تلاش پیش بینی شده برای پشتیبانی

گام اول کلاس‌های تغییرات مورد انتظار را بر مبنای برنامه‌کاربردی یا توصیف برنامه فرموله می‌کند، سپس برای هرکدام از وظایف مربوط به پشتیبانی یک مجموعه نمایان‌گر از سناریوها تعریف می‌شود. به سناریوها یک وزنی برمبنای احتمال رخ دادشان در یک فاصله زمانی عملی نسبت داده می‌شود. برای این‌که قادر به ارزیابی سایز تغییرات باشیم، سایز همه کامپوننت‌های سیستم مشخص می‌شود. یکی از سه تکنیک می‌تواند برای تخمین سایز کامپوننت‌ها استفاده شود: به‌کارگیری تکنیک تخمین انتخاب، یک سازگاری از یک متریک شی گرایی یا زمانی‌که داده‌ای از برنامه‌های کاربردی مشابه قابل دسترس است، سایز داده موجود می‌تواند استفاده شود و برای کامپوننت‌های جدید قیاس شود. کل تلاش مورد نیاز جهت پشتیبانی حاصل‌ضرب پیچیدگی سناریوها در احتمال‌شان پیش بینی می‌شود.



شکل ۷: ورودی‌ها و نتیجه ALPSM

● **استفاده مجدد از دانش پایه:**

مورد بررسی قرار داده نمی‌شود، اما داده‌های به جا مانده از برنامه‌های کاربردی مشابه یا گونه‌های اولیه محصول مورد نیاز می‌شود. داده‌های اولیه به‌کامپوننت‌های جدید قیاس می‌شود.

۸- A Software Architecture Evaluation Model (SAEM)

فرآیند ارزیابی نیازمندی‌های کیفیتی SA به سختی فرموله می‌شود، به‌ویژه در مورد متریک‌های توصیف شده در مدل موجود در یک مدل کیفیتی بر مبنای Standard Software Quality Assessment Process انتخاب می‌شود و یک چهارچوب مفهومی‌که نیازمندی‌های کیفیتی، متریک‌ها و خصوصیات داخلی SA و سیستم نهایی را مربوط می‌کند، پیشنهاد می‌شود. عناصری برای ارزیابی کیفیت یک سیستم نرم‌افزاری برمبنای استاندارد خاصی مورد نیاز است که عبارتند از: یک مدل کیفیتی، یک متد برای ارزیابی و ابزار پشتیبانی.

● **هدف ویژه متد:**

SAEM تلاشی در جهت تعریف متریک‌های کیفیتی برمبنای تکنیک Goal-Question-Metric (GQM) انجام می‌دهد.

● **مشخصه‌های کیفیتی:**
کیفیت به دو دسته خارجی و داخلی تقسیم می‌شود. کیفیت خارجی دیدگاه کاربر و کیفیت داخلی دیدگاه توسعه دهنده را بیان می‌کند. برقراری ارتباط بین صفات داخلی و مقادیرشان امری ضروری است؛ QFD به عنوان یک تکنیک مناسب برای این هدف توصیه می‌شود.

● **افراد درگیر:**
دانش افراد خبره و داده انباشته شده مربوط به شرکت در نگاشت نیازمندی های کیفیتی به صفات داخلی استفاده می‌شود.

● **توصیف SA:**
SA از دو دیدگاه متفاوت مورد بررسی قرار داده می‌شود، یکی از دیدگاه توسعه دهنده و دیگری از دیدگاه کاربر می‌باشد. بنابراین، SA یک محصول نهایی یا یک محصول میانی در فرآیند توسعه سیستم نرم‌افزاری می‌باشد. فرآیند توسعه معماری صفات داخلی را محدود می‌کند، بنابراین نتیجه فرآیند اندازه‌گیری می‌تواند معماری را در یک فرم از بازخور اصلاح کند.

● **فعالیت‌های متد:**
SAEM یک مدلی برای ارزیابی کیفیت برمبنای مجموعه داده، اندازه‌گیری و تحلیل نتایج می‌دهد. فرآیند تحلیل به دو بخش فرآیندهای داخلی و خارجی تقسیم می‌شود و با دیدگاه‌های کاربران و توسعه دهندگان سازگار می‌شود. نیازمندی های کیفیتی مشخص شده برمبنای دانش افراد خبره و داده ذخیره شده در شرکت به خصوصیات داخلی نگاشته می‌شوند که روی SA ارائه خواهند شد.
استفاده مجدد از دانش پایه‌ای موجود، مورد بررسی قرار داده نشده است. هرچند، مدل ارزیابی وجود یک کیفیت داخلی اولیه را فرض می‌کند، کیفیتی که صفات داخلی مورد انتظار را با مقادیرشان و روال ارزیابی شان تعریف می‌کند.
SAEM روی هر سیستم نرم‌افزاری معتبر شناخته نشده است.