

# سند ۴+۱ در معماری نرم افزار

علیرضا سلطانی نشان

۱۱ آبان ۱۴۰۳

در سال ۱۹۹۵ آقای Philippe B. Kruchten مقاله‌ای را تحت عنوان Software Architecture منتشر کرد که در بند اول آن نوشته بود [۱]:

معماری نرم افزار در سطح بسیار بالایی از تجرید قرار دارد و طراحی منطقی یک نرم افزار می باشد در حالی که طراحی نرم افزار کمترین مقدار تجرید را دارد و تمام جزئیات المان ها و ارتباطاتشان با یکدیگر مشخص است و همچنین شامل طراحی فیزیکی نیز می باشد. آقای Kruchten بر این باور است که نوشتن ایده ها بر روی کاغذ همیشه مفید خواهد بود قبل از اینکه آن را کد کنیم. ایشون به صراحت مطرح می کند که همیشه مصورسازی بهتر از خواندن مستندات کسل کننده است و در نهایت مهم ترین انگیزه برای تهیه سند ۴+۱ از نظر او آن است که مدل سازی آن آسان است زیرا باعث می شود که سیستم را از دیدگاه های مختلفی بررسی کنیم. ایده مدل معماری ۴+۱ برای رفع مشکل جمع شدن اطلاعات بیش از حد در یک نمودار معماری نرم افزار یا عدم رسیدگی به برخی از نگرانی های ذینفعان بوده است تا بتواند تمام جنبه ها را در ۴+۱ مدل در یک معماری نرم افزاری پوشش دهد.

جدول ۱: تفاوت میان طراحی و معماری در نرم افزار [۲]

معماری	طراحی
در سطح کل سیستم ویژگی های اصلی و اساسی سخت در ایجاد تغییرات به دلیل تاثیر در کل سیستم	در سطح هر ماژول و کامپوننت ویژگی های جزئی در سیستم آسان در ایجاد تغییر به دلیل Separation of Concerns

## ویوهای سند ۴+۱

مدل ۴+۱ View توصیفی از معماری نرم افزار را با استفاده از پنج View همزمان مدیریت و سازماندهی می کند که هر کدام از View ها به مجموعه خاصی از نگرانی ها می پردازد:

۱. Logical View: تمام عملیاتی که سیستم می تواند به کاربران نهایی ارائه دهد را در این قسمت مدیریت خواهیم کرد که عموماً از نیازمندی های عملیاتی یا Functional requirements تشکیل شده است و از نمودارهای کلاس و کامپوننت می توانیم در آن کاربران و مدل هایشان را تعریف کنیم.

۲. Process View: تمرکز اصلی آن بر تمام نیازمندی های غیرعملیاتی یا Non-functional requirements در زمان اجرای سیستم است که شامل موارد، امنیت، کارایی بالا، در دسترس بودن، مدیریت همزمانی، و تحمل خطا<sup>۱</sup> می باشد و از نمودارهای Sequence و Activity برای نمایش آن ها استفاده می کنیم.

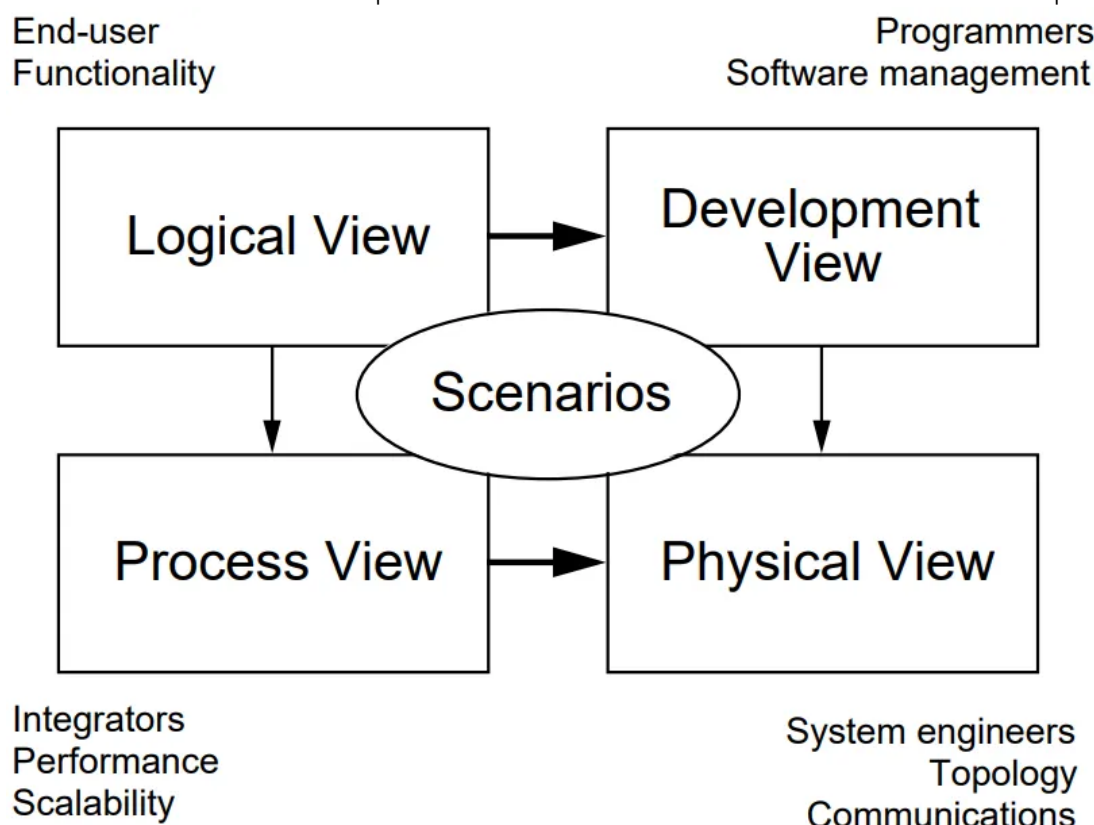
۳. Development View: این ویو سیستم را از دیدگاه یک توسعه دهنده نشان می دهد و با مدیریت نرم افزار سر و کار دارد. عموماً در این ویو از نمودارهای پکیج و کامپوننت استفاده می شود.

۴. Physical View: توصیف نمود نرم افزار بر روی سخت افزار و ارتباطات فیزیکی بین کامپوننت ها را مطرح می کند. عموماً از نمودار Deployment برای نمایش استفاده می شود.

۵. Scenarios: توصیف رفتار سیستم از نظر یک کاربران نهایی ای که از این سیستم می خواهد استفاده کند را نشان می دهد که از نمودار Usecase استفاده می کند.

<sup>۱</sup>Fault-tolerance

شکل ۱: چهار ویو مورد استفاده برای نشان دادن و اعتبارسنجی نیازمندی‌های یک پروژه نرم‌افزاری با یکدیگر پیوند دارند که در ابتدا بایستی آن‌ها را ایجاد کنیم و سپس از نمودارهای مناسب برای مصورسازی آن‌ها استفاده کنیم. [۱]



## مزایا و معایب

سند ۴+۱ به عنوان یک View model دارای مزایا و معایبی زیر است [۳].

### مزایا

- یک ساختار صریح و سازگار را به سازمان‌ها ارائه می‌دهد که به آن‌ها کمک می‌کند تا اطلاعات خود را مرتب و سازمان‌دهی کنند.
- این سند تمام دیدگاه‌های نسبت به سیستم را پوشش می‌دهد مانند نیازمندی‌های غیر-عملیاتی زیر:
  - Functionality
  - Performance
  - Reliability
  - Scalability
  - Security
  - Deployment
- از سطوح چندگانه تجرید و جزئیات پشتیبانی می‌کند و به ذینفعان کمک می‌کند که روی View که بیشتر ارتباط را به آن‌ها دارد تمرکز کنند.
- ارتباطات و همکاری بین تیم توسعه، مشتریان، مدیران، و تسترها را از نظر استفاده از اصطلاحات<sup>۲</sup> و Notation<sup>۲</sup>ها بهتر می‌کند تا فهم بیشتری در گفت و گوها داخلی بین گروه‌های مختلف بدست آید.

<sup>۲</sup>Terminology

- این سند ممکن است با تمام سیستم‌ها یا دامنه‌ها، مناسب و Fit نباشد. ممکن است به خاطر بعضی از جنبه‌های سیستم نیاز باشد که View‌های مناسب نسبت به آن طراحی و دوباره تعریف شوند.
- به دلیل تغییرات سریع در سیستم مورد نظر در گذر زمان، ممکن است این سند وضعیت کنونی سیستم را نشان ندهد و برخی از قسمت‌های آن منسوخ شده و ناسازگار با سیستم جدید (در حال توسعه) باشد.
- ممکن است تمام تصمیمات طراحی و سبک سنگین‌هایی که<sup>۳</sup> در طول فرایند توسعه رخ می‌دهد را در بر نگیرد چرا که ممکن است بعضی از آن‌ها به صورت ضمنی در جلسات باشند و حتی مستندسازی نسبت به آن انجام نشده باشد.
- این سند تلاش بسیار زیادی را برای مدیریت نیاز دارد تا بتواند ۵ لایه View را نسبت به هم هماهنگ و منسجم نگه دارد به همین خاطر نگهداری و به روز رسانی آن کار آسانی نیست.

## بررسی نهایی نمودارهای مورد استفاده

جدول ۲: نمودارهایی که می‌توان در سند ۴+۱ متناسب با ویو استفاده کرد [۲].

نام ویو	دیاگرام UML
Logical view	Component diagram, Object diagram, Class diagram Composite Structure Diagram, Package diagram
Process view	Timing diagram, Sequence diagram, Activity diagram Interaction overview diagram, State machine diagram
Development view	Component diagram, Package diagram
Physical view	Deployment diagram
Scenarios	Usecase diagram

## جایگزین‌ها

اسنادی که می‌توانند به صورت جایگزین نسبت به سند ۴+۱ استفاده شوند به ترتیب زیر هستند:

۱. سند C4
۲. سند TOGAF
۳. سند 5+2
۴. سند SysML
۵. سند ArchMate
۶. سند ADL

## مراجع

- [1] Kruntchen, P. Architectural blueprints—the “4+1” view model of software architecture. *IEEE software*, 12(6):42–50, 1995.
- [2] ), ( . 4+1 view model of architecture. <https://medium.com/2b-1-better-2day/4-1-view-model-of-architecture-1a1794ff8110>, 2024.
- [3] Community. What are the benefits and drawbacks of using the 4+1 view model for software architecture documentation? <https://www.linkedin.com/advice/3/what-benefits-drawbacks-using-41-view-model>, 2024.