

اهداف طراحی زیرساخت

- کاهش هزینه و زمان پیاده سازی (Cost & Time)
 - با استفاده از قانون 80-20
 - Compiler Error بجای Run Time Error
- افزایش کیفیت (Quality management)
 - Compiler Error بجای Run Time Error
 - کاهش حجم کدنویسی
- افزایش Maintenance
 - Change Management
 - Code Management
- سایر مزیت های این زیرساخت
 - هیچ نیازی به برنامه نویسی Java Script و یا Type Script وجود ندارد.
 - نیاز به برنامه نویسی در سمت کلاینت و سرور بصورت مجزا وجود ندارد.

کاهش هزینه و زمان پیاده سازی: در اینجا می خواهیم با استفاده از **قانون 20-80** ثابت کنیم که زیرساخت باعث کاهش هزینه و زمان پیاده سازی تا **حدود یک چهارم** می شود.

قانون 20-80: این قانون یک قانون کلی است که در بسیاری از علوم مورد استفاده قرار می گیرد، براساس این قانون 80 درصد یک نرم افزار ماهیت یکسان و 20 درصد آن ماهیت متفاوتی دارد.

- **بدون استفاده از زیرساخت:** بدون استفاده از زیرساخت هر بخش به اندازه ی همان بخش هزینه و زمان دربرمی گیرد. بنابراین اگر هزینه و زمان بخشی که ماهیت متفاوت دارد را X بگیریم، هزینه و زمان بخش دیگر برابر $4X$ (20 در برابر 80) و هزینه و زمان کل برابر $5X$ ($X + 4X$) می شود.
- **با استفاده از زیرساخت:** با استفاده از زیرساخت **80 درصد** زمان صرف 80 درصد دیگر می شود، در هر دو حالت بخشی که ماهیت متفاوتی دارد دارای هزینه و زمان یکسانی است (چون زیرساخت برای بخشی که ماهیت متفاوتی دارد کارایی ندارد) همانند قبل بخشی که دارای ماهیت متفاوتی است را برابر X می گیریم، هزینه و زمان بخش دیگر برابر $X/4$ (80 در برابر 20) و هزینه و زمان کل برابر $5X/4$ ($X + X/4$) می شود. بنابراین هزینه و زمان به $1/4$ حالت قبل کاهش می یابد.



Cost & Time



Cost & Time



- **Compiler Error بجای Run Time Error:** این زیرساخت بگونه ای طراحی شده که بسیاری از خطاها در زمان کامپایل قابل شناسایی هستند، و براساس یک تحقیق برطرف کردن یک خطا در زمان کامپایل حدود 12 بار سریعتر از برطرف کردن همان خطا در زمان اجرا است.

افزایش کیفیت: در اینجا مشخص می کنیم که استفاده از زیرساخت چگونه باعث کاهش خطا و افزایش کیفیت نرم افزار می شود.

- **Compiler Error بجای Run Time Error:** همانگونه که قبلا نیز گفته شد این زیرساخت بگونه ای طراحی شده که بیشتر خطاها در زمان کامپایل قابل شناسایی هستند، و این امر علاوه بر کاهش هزینه و زمان باعث کاهش خطاها و افزایش کیفیت نرم افزار می شود.
- **کاهش حجم کد نویسی:** بصورت بدیهی کاهش حجم کدنویسی باعث کاهش خطاها می شود، همچنین استفاده مجدد از کلاسها و کامپوننت هایی که قبلا مورد استفاده قرار گرفته اند احتمال خطا را بشدت کاهش می دهد.