

# عقایدی درباره طراحی UI

این مقاله چندین قانون طراحی UI را توضیح می دهد. به اعتقاد بنده تبعیت از این قوانین آغاز خوبی برای یک برنامه کاربردی با UI بهتر خواهد بود.

## مقدمه

در دوران ما، اهمیت طراحی یک UI خوب کاملاً شناخته شده است. نسبت به زمان برنامه های خسته کننده کنسول، ما پیشرفت های زیادی کرده ایم. امروزه ما تنها در مورد طراحی شهودی UI صحبت نمی کنیم، بلکه صحبت درباره برنامه های client هوشمند با قابلیت های گسترده است. طراحی برنامه کاربردی UI خوب به نظر کار آسانی است، ولی در واقع اصلاً این طور نیست. ملاحظه نیازهای کاربران که مسلماً بیش از صدها کاربر هستند، کار سختی است.

## چه کسی باید این مقاله را بخواند؟

این مقاله احتمالاً برای طراحان UI، مدیران پروژه یا تولید، و البته توسعه دهندگان مفید است.

## ترک دعوی

این مقاله شامل عقاید شخصی من درباره چگونگی برخورد با طراحی UI است. تصور می کنم عده ای از کاربران با بعضی از این قوانین مخالف هستند، و این خوب است. روی هم رفته، وسعت کیفیت UI در چشمان ناظر است. با این وجود، این مقاله منتشر می شود زیرا قوانینی که در آن تشریح شده با اکثر برنامه های UI ویندوزی که تا به حال استفاده کرده ام، مطابقت می کند.

## سازگاری

اگر یک UI فاقد سازگاری باشد، از بین می رود. سازگاری هم باید در ظاهر و هم در کاربرد باشد. اگر تا به حال با Microsoft Outlook کار کرده باشید، احتمالاً می دانید در چه مورد صحبت می کنم. تا به حال چند بار از میانبر Ctrl + F در Outlook استفاده کرده اید و فهمیدید که این میانبر در این برنامه ایمیلی را forward می کند؟ این فقدان سازگاری موجب مخالفت من شده و هر بار نیز تکرار می شود. این یک اشکال است. این اشکال، علاوه بر اینکه باعث شده این دوگانگی در Redmond کاربران را آزار دهد، موجب بدنامی ماکروسافت نیز شده است. از زمان پدیدار شدن سیستم عامل ویندوز، ماکروسافت تلاش کرده تا توسعه دهندگان را به نوشتن برنامه های کاربردی ویندوز با UI سازگار، ترغیب کند.

استفاده از WinAPI همواره رایگان بوده و هست (در مقابل چیزی که در MAC وجود دارد). آنها framework های نظیر MFC، VB، ATL و ابزارها را توسعه داده اند. علاوه بر این ماکروسافت از توسعه دهندگان می خواست تا سازگاری در شرایط استفاده از UI ویندوز را حفظ کنند، اما این در Outlook شکست خورد.

مثال ۱:



در مثال بالا، طراحان UI نرم افزار Nero تصمیم گرفته اند تا به وسیله طراحی یک پنجره غیر سنتی، نوآوری کنند. با این که سریع به نظر می رسد، اما هنوز گیج کننده است زیرا دکمه خروج در سمت راست بالای پنجره قرار ندارد، بلکه در پایین است. آیکن خروج نیز به هر دلیلی شبیه آیکن خاموش کردن در ویندوز XP است.

برای انجام سازگاری UI، باید رفتارهای برنامه های متعارف ویندوز را دنبال کنیم. برنامه منتخب من می تواند یکی از برنامه های بسته نرم افزاری Office باشد. من همواره به عنوان مرجع به برنامه های بسته Office اشاره می کنم، زیرا:

اولاً طراحی UI این بسته به خوبی خطاگیری شده است. ثانیاً زیرا کاربران برنامه من به شدت از آنها استفاده می کنند. ثالثاً ماکروسافت هزینه و فکر و تلاش بسیاری را صرف طراحی آن کرده است.

یکی از همکاران من از این حقیقت که او نیاز دارد تا Office را مانند دیگر برنامه ها توسعه دهد، متفر دارد. او سابقاً از اینکه خلاقیتش در این ورطه غرق می شود، شکایت می کرد. و من به او می گفتم که اصلاً مهم نیست که او چقدر از چیزی که Office UI ماکروسافت ارائه می کند ، متفر است. هنوز دلایل خوبی برای استفاده از آن وجود دارد و این فقط به خاطر کاربر نهایی نیست. تقلید از نوار ابزار و menu های Office کار توسعه را آسان می کند زیرا از تجربه ماکروسافت در طراحی UI استفاده می کنیم. اگر شرکتی با چنین ظرفیتی، پول و وقت زیادی را بر روی این موضوع صرف کرده، چرا باید دوباره این هزینه صرف شود ؟

مثال ۲:



در مثال بالا می توانیم ببینیم که میانبر ALT مثلاً برای منوی File وجود ندارد. آیکن ها در هر طرح بندی به نظر می رسد که توسط طراح گرافیکی متفاوتی طراحی شده اند. آیکن Export معنایی ندارد. صفحه آرایشی بسیار سنگین است.

در جاهایی که ویندوز سازگار نیست یا شما می خواهید طراحیتان متفاوت باشد، باید مجموعه ای از قوانینی که در تمام برنامه (های) شما باید رعایت شوند را بنویسید. مثلاً اگر شما از نوار ابزار شخصی استفاده می کنید، در همه فرم ها آن را نگه دارید. یک دفعه تصمیم نگیرید که آن را در یکی از فرم هایتان حذف کنید. مثال دیگر، استفاده از منوی محتویات است. اگر شما آیتم های خاصی در منوی محتویات دارید، مثلاً: Copy، Cut و Paste باید آن ها را در همه فرم های وابسته و با ترتیب یکسان حفظ کنید. ممکن است استفاده آن ها متفاوت باشد، یا منوهای محتویات شامل آیتم های خاص اضافه تری در هر کنترل باشند، اما به منظور حفظ سازگاری، آیتم های معروف باید باقی بمانند. همچنین بسیار مهم است که از خانواده فونت و اندازه یکسانی استفاده کنید و صفحه بندی مشابهی را در پنجره ها حفظ کنید.

## همواره در مورد کاربران نهایی فکر کنید

این قانون به مزاح می تواند اینگونه باشد: "واقعاً روزی کسی از این استفاده خواهد کرد؟" تنها چیزی که هست این که چیزی برای خندیدن به آن وجود ندارد.

این قانون خیلی مهمی است. اگر شما در هر مرحله از طراحی UI و در مراحل توسعه به کاربر فکر نکنید، آنگاه هیچ مشتری ای، حتی راضی ترین آنها، از برنامه شما استفاده نخواهد کرد. توسعه دهندگان و طراحان (که خود سابقاً توسعه دهنده بوده اند) متمایلند که مهارت های کاربران نهایی را فراموش کنند.

آنها خود را به عنوان کاربر نهایی تصور می کنند. در واقع، بسیاری از کاربران نهایی مهارت کمتری نسبت به توسعه دهندگان دارند.

این قانون به سادگی برآورده نمی شود، زیرا روز به روز، در جدول طراحی و توسعه، با نابهنجاری هایی آشنا می شویم و طراحی بد UI احساس می شود.

تشخیص نوع کاربری که از برنامه استفاده خواهد کرد، بسیار مهم است. اغلب این بهترین کار است که روی پایین ترین قسمت مشترک کار کنیم و قابلیت های بیشتری را در آن پنهان کنیم. این کار برای اینکه به هر کاربر اجازه دهد تا برنامه را بسته به دانایی و توانایی و تخصص خودش تغییر دهد، فوق العاده است. این تغییرات داده ها باید برای هر کاربر در یک رسانه ماندگار ذخیره شود، و بعد از به روزرسانی برنامه همچنان دست نخورده باقی بماند.

به نظر من، کیفیت تجربه کاربران توسط معیارهای زیر تعیین می شود:

♦ آیا ویژگی ها و کارکردهایی که من می خواهم، وجود دارد و به خوبی کار می کند؟

♦ آیا برنامه ساده، شهودی و آسان است؟

♦ آیا برنامه از نظر جنبه های کارایی ، سریع است ؟

♦ آیا برنامه ثبات دارد ؟

اگر کاربر به سوالات فوق پاسخ مثبت دهد ، آنگاه با یک کاربر راضی که بارها و بارها با رضایت از برنامه استفاده خواهد کرد، روبه رو هستیم.

نیازهای کاربران را در نظر بگیرید تا آنها نرم افزار شما را تحسین کنند ، در غیر این صورت شما را به خاطر تمام مشکلات دنیا سرزنش خواهند کرد.

مشکل اصلی زمانی است که انواع مختلفی از کاربران با انتظارات مختلف وجود دارند. در یکی از پروژه هایی که من در آن شرکت داشتم ، متوجه چیزهای خیلی جالبی در رابطه با استفاده از UI شدم. UI برنامه در سه جای مختلف با کارکرد یکسان ظاهر شده است: منو و نوار ابزار و منوی محتویات. در ابتدا این مانند موضوع افزونگی بود، ولی خیلی زود دیدم که سه نفر متفاوت در شرکت من بر اساس ارجحیت خود ، از یکی از این گزینه ها استفاده می کنند. استفاده از این راهکار ، انواع مختلف کاربران را راضی می کند. تکنیک دیگری که کمک می کند تا در طول پروسه توسعه و طراحی به فکر کاربر نهایی باشیم ، این است که بر اساس مبنای معینی با آنها ملاقات کنیم و در مراحل پیش فروش بازخورد آنها را جمع آوری کنیم. همچنین شما باید بعد از دوره تست و فروش ، نظرخواهی ای در بین مشتریان انجام دهید. نظرخواهی از کاربران باعث می شود تا آنها احساس کنند که شما به کیفیت کار اهمیت می دهید و موجب می شود آنها صبورتر شوند و در رابطه با اشکالات برنامه با شما همکاری کنند.

### برآورد دست بالا در مورد توانایی های کاربر نکنید و آن را درست انجام دهید

طراحی UI خوب در برنامه ها فکر کردن در رابطه با چگونگی ساخت نوار ابزار نیست، بلکه تعیین نوعی که باید باشد، است : آیا باید dockable ( قابل قطع ) باشد یا نه؟ چه چیزهایی باید در آن قرار داد؟ آیا به آن نیاز است؟

کاربر ممکن است هنگام کار با برنامه شما سوالات زیر را بپرسد : چگونه یک تکلیف انجام می شود؟ کجا انجام می شود؟ چه وقت انجام می شود؟

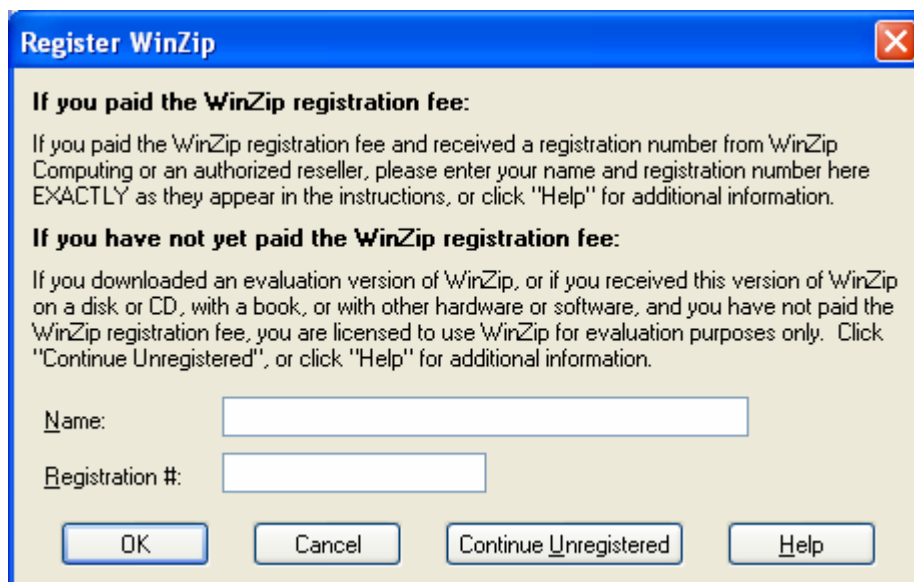
مطمئن شوید که کاربر می تواند در مدت زمان کوتاهی به این سوالات پاسخ دهد. هر چه در مدت کوتاه تری این انحنایا درک شوند ، کاربران راضی تر خواهند بود.

♦ از سخت بودن پیام ها و کنترل دیالوگ هایی که نیاز به تصمیم کاربر دارند ، بپرهیزید. حداقل باید

تا جایی که می توانید آن را واضح تر کنید. کاربران از خواندن پیام هایی نظیر : " .. فلان و بسار ...

اتفاق افتاد ... " و بعد از چندین خط وحشت آور از آنها خواسته شود تا به پرسشی مثل " آیا می خواهید عمل X را انجام دهید؟ " جواب دهند ، متتفرند.

کاربر در مورد این گونه پیام ها به شما فحش نمی دهد، پس او را پر از اطلاعات نکنید. برای کاربر شما کار سختی که در مورد چه چیزهایی تصمیم بگیرد و یا نگیرد. مرز ناچیزی بین آزار و آشفستگی کاربران و اجازه آشنایی با مسائل را به آنها ندادن، وجود دارد.



در مثال بالا شما می توانید استفاده شدید از کلمات را در واسط WinZip ببینید. وقتی این پیام را می بینید ، با خود می گوئید " اوه .. من ۱۵ دقیقه وقت برای خواندن همه این موارد دارم ؟ "

♦ وابسته به دستورالعمل و فایل های help نباشید. احتمالاً در مورد RTFM شنیده اید (Read The F\*\*\*ing Manual). این یک شرمندگی بزرگ است ، اما واقعیت دارد که کاربران اصلاً این فایل ها یا دستورالعمل ها را نمی خوانند یا حداقل سعی می کنند هر کاری انجام دهند تا از زیر بار خواندن آن ها فرار کنند. هر چه قیمت برنامه بیشتر باشد، احتمال اینکه کاربران دستورالعمل آن را بخوانند بیشتر می شود.

به عنوان مثال اگر برنامه رایگان باشد و کاربر هم نتواند خوب با آن کنار بیاید ، احتمالاً آن را uninstall می کند. حقیقت فوق به این معنا نیست که می توانید از نوشتن دستورالعمل طفره بروید؛ چون مشتری هنوز به آن نیاز دارد. کاری که می توانید انجام دهید این است که UI را برای استفاده، تمییز و واضح و ساده کنید.

♦ هرگز چنین تصور نکنید که کاربر شما به قدری باهوش است که رنگ زمینه را مشکی نمی کند زیرا رنگ فونت آن مشکی است یا چون ظاهر آن زشت می شود.

از قانون اصلی Murphy استفاده می کنم و می گویم اگر UI برنامه به کاربر اجازه دهد که لنگش را هوا کند، او احتمالاً این کار را می کند. مطمئن شوید که برنامه در مقابل کاربران احمق محافظت شده، در غیر این صورت با سرزنش های شرکت های احمق سروکار خواهید داشت.

♦ هرگز پیام های خطایی را که کاربر نمی تواند آن ها را هضم کند، به کاربر تحمیل نکنید. پیام های اطلاعاتی UI را کوتاه و توضیحی نگه دارید. به جای آن ، log های غنی برای ارسال به قسمت پشتیبانی مشتری ، ایجاد کنید. قبل از ارسال برنامه، مطمئن شوید که پیامهای ترسناکی مثل "... مشکل وخیم ..." یا "... ما نباید به چنین کدی برسیم ..." را واضح و درست کرده اید.

مطمئن شوید که برنامه زیر بار هر رویدادی دوام می آورد. تمام خطاها را پیش بینی کنید و تنها پیامهای کاربر پسند را نمایش دهید.

♦ نپرسید که کاربر در جهت استفاده بهتر از برنامه شما، چه کاری می تواند بکند و چطور حداکثر استفاده را می برد. بپرسید که برنامه چه کاری می تواند برای کاربر انجام دهد.

کاربران نهایی تنبل هستند. اگر کار آنها را راحت کنید، از شما ممنون خواهند شد. ابزاری به کاربران بدهید تا آنها بتوانند فرآیند های نیمه اتوماتیک یا تمام اتوماتیک ایجاد کنند تا مجبور نباشند کارهای مشابه را بارها و بارها انجام دهند و وقت خالی برای خواندن آخرین اخبار روی وب داشته باشند. البته دارم در مورد شما صحبت می کنم.

شما یک استثنا هستید. من در مورد Dilbert و Homer Simpson صحبت می کنم!

اگر منابعی برای ساده سازی هر قسمت از برنامه تان را ندارید، حداقل مطمئن شوید که روند اصلی و عملیتهای مکرر را تا آن جا که امکان داشتند، ساده کرده اید.

## شخصی سازی

هیچ چیزی به عنوان اغراق در مورد قابلیت های شخصی سازی وجود ندارد. تا به حال نشنیده ام که کسی از برنامه ای به علت اینکه گزینه های زیادی برای شکل بندی دارد، شکایت کند. فقط ممکن است از اینکه این گزینه ها خوب تنظیم نشده اند، عصبانی باشد. این قانون بسیار مهم است، زیرا: مقدار گزینه های شکل بندی ای که یک برنامه دارد با راحتی تناسب دادن برنامه برای مشتری ای خاص\_ کاری که معمولاً برای اعضای تیم پروژه و مشتری پر زحمت است\_ در یک راستاست.

وسعت زیاد گزینه های شخصی سازی به کاربرانی که در سطوح متفاوت مهارت هستند اجازه خواهد داد تا برنامه را با نیازهایشان وفق دهند. مسلماً باید داده های شخصی سازی برای اجراهای بعدی برنامه، نگهداری شوند.

برنامه ای که قابلیت شخصی سازی خوبی دارد، استفاده از برنامه را برای افراد معلول ممکن می سازد. مثلاً ممکن است عده ای از کاربران نهایی کوررنگی داشته باشند. در این مورد، وجود گزینه تغییر رنگ برنامه الزامی است.

## ماشین کاربر را در نظر بگیرید

هنگامی که برای آیکن های نوار ابزار از کیفیت رنگ ۳۲ بیتی استفاده می کنید، کاربرانی را که از کیفیت رنگ بیش از این استفاده می کنند، نادیده گرفته اید. صفحه نمایش با رنگ ۱۶ بیتی. فاکتور دیگر، اندازه صفحه نمایش است. اگر این دسته از ملاحظات کنترل نشوند، برنامه شما ضعیف به نظر می رسد.

## ماشین کاربر را در نظر بگیرید

UI خوبی که قوانین فوق را رعایت کرده باشد، به شرکت کمک می کند تا محصولاتش را بفروشد. گاهی شرکت های رقیب شانس های تجاری را از شما می دزدند، فقط به خاطر اینکه UI آنها بهتر به نظر می رسد. یک UI خوب، می درخشد و تأثیر خوبی می گذارد. اصلاً مهم نیست که برنامه شما کارکرد قوی تری دارد. "اگر تکنیک برنامه پایین به نظر برسد، هیچ شانسی برایش وجود ندارد."

این مقاله تنها قطره ای از دریای نکته ها و قوانین UI را توضیح داد. شما را به خواندن و یادگرفتن در این موضوع تشویق می کنم، این تنها راه بهتری نوشتن UI است.