

۱- ۷ اصل تعامل انسان و کامپیوتر را شرح نمائید ، سپس برای هر یک از آنها مثال مناسب ، خارج از نمونه های کتاب با توضیح مناسب ارائه نمائید .

۱- کاربر را شناس : نخستین عقیده در HCI ، ایجاد ارتباط و تعامل حول محور کاربر مورد نظر است . این عبارت توسط Hansen در ۱۹۷۱ ساخته شد . این عبارت تا چند سال افق دیدی را نشان می داد . این اصل به یادگاری بیان می کند که تعامل در رابطه باید نیازها و ویژگی های کاربر استفاده کننده از سیستم را فراموش نکند . با اینکه این اصل ممکن است ساده انگارانه باشد ، اغلب طراحان HCI و ارگانون ها بدون درک کامل از کاربر حل می دهند . به طور ایده آل ، اطلاعات و افصح (مانند سن ، جنسیت ، سطح آموزش ، وضعیت اجتماعی یا شغلی ، تجربه کامپیوتری ، پیشینه فرهنگی) درباره هدف مورد نظر ، بودجه آمی و ارزش های مورد توجه یا ترجیحات بهینه ، نیازات ، توانایی ها و شیوه های درک و سطح مهارتشان در تصمیم گیری فرضا مثال یک توسعه دهنده برای تصدیق کار می کند . (برای انتقاد بهتر) او باید بداند که عنوان کم سن و خردسال میسر می شود یا نه و در این صورت باید می کند . به عکس سالمنان . به اعتقالات قدیمی فعلی نمی توان گفت که چون ممکن است اگر یک رشته تحقیق در دسترس نیست ، یک طراح HCI کار کرده ، با رفتار حداقل تلاش می کند تا اطلاعات گسترده را شناسایی ، الگوریتمیک و داده های شده انسان را نسبت بندی کند تا به دیگران و خفیه های گروه کاربر هدف برسد که عمده و عینی از کارهای طراحی می شود . مستقر ، بحث استفاده جهانی است که به سختی تعاملات انسان را و سطوح ناتوانی . هر حال ، انجام این محدوده عظیم دیگر تعامل امکان پذیر نیست . مثلا استفاده جهانی بحث سرمایه گذاری را به چالش می کشد که تعاملات جهانی را برای گروه های خاص جداگانه طراحی می کند

۲- Task را شناس : گویا ایدهی تئریایی یا بیانی - منطقی ، تا اندازه ای اصل طراحی HCI بر مبنای شناخت

Task است . این اصطلاح رواج می کند به کاری که توسط کاربر در طی سیستم تعاملی انجام می دهد حقیقت شناخت Task فعلی ارتباطات و تدبیر زیادی به مدلسازی تعامل و ارزش های کاربر دارد . این واقعاً شناخت ارتباط و ساختمان کاربرها در یک سطح انتزاعی مناسب برای کاربرها و دیگر ابلیشن محاوره ای ضرورت می کند . توجه داشته باشید که Task مدل باید به طور ایده آل از دست کاربر باشد . کاربران متفاوت ، مدل های ذهنی متفاوت از کارمندی خواهند داشت و این باید به ساختار تعامل یاداب شود تا برای آن برای تمامی کاربرها راحت تر شود . بهر حال این همیشه مورد نیست که مدلسازی تعامل بعد از کاربر بهترین روش باشد . گویا باید به یاد داشته باشید که آنها بسیار دقیق شوند و حتماً عملی که یک مدل غیر کاربر - محور Task را عمل ممکن است بر مبنای ظرفیت عادی انسان و نهایی توسعه داده شود .

۳- کاهش از بار حافظه : سخت اطرا می یک عامل با حداقل وزن حافظه ندارد بطور تنوع قابل انجام است.

انسان ها قطعا بطور طبیعی کارهایی که بار حافظه ندارد انجام می دهند ، طولانی یا کوتاه مدت .
کم کم داشتن بار حافظه کوتاه مدت کاربر از اصل ترین اهداف ها در کنار نقش قابل به طور سریع و راضیانه راه برای
انجام Task ها است . طریقت حافظه کوتاه مدت انسان ، حدود ۵ - ۹ قطعه اطلاعات است مشهور به عدد جادویی
بارکم حافظه همچنین به رفتارهای کم حافظه می انجامد .

۴- تلاش برای ثبات داشتن : در بحث طولانی مدت (حافظه) یک راه برای کم کردن بار حافظه ، ثبات داشتن است.

این برای یک اپلیکیشن و در طول اپلیکیشن های متفاوت و مردم مدل عامل و اطری ارتباطات مدتی می کند
توجه داشته باشید که همان زیر کار (Sub Task) ممکن است چند بار در طول اپلیکیشن های متفاوت ظاهر شود .
در کنار بار داشتن اسکید می باید کرد ، ثبات و آشنایی همچنین راه می بود به مقبولیت بیشتر و تریج .

۵- یادآوری به کاربر و تار کردن ذهن ایشان : در Task خاص شامل استفاده از حافظه است ، پس یک استوری

خوب و دیگر ، استخدام عامل های است که به یاد آوری های بیایی درباره اطلاعات مهم می دهند و در نتیجه حافظه را
تازه می کنند . حافظه انسان اطلاعات را به سرعت از هم دور می کند و این به خصوص در مواردی درست است که
سین چندین کار ، چند کاری می کنند . در حقیقت تحقیقات نشان می دهند مقدار به طور داخلی ، که گذر می کردن کارها را در طول
چند کاری انجام می دهد . برای مدت آوردن هویتیاری کاربر درباره شرایط و استخراج با پیخ های دست جلوتر ، با محتوا ،
نقطه ای در یای بیایی ، باز فرود ذهن کاربر با تازه فواید گردیده بود که می کنند Task را راحت تر انجام دهد .
یکی از راه های مخصوص باز فرود نامکتوا ، باز بیان عملیات کاربر است برای سلیان ذهن به خاطر دادن یک

میرود می حجم . این خاتمه یک اطمینان خاطر توسط جوهر کردن تصویر ذهنی کاربر درباره کاردهای انجام می دهد

۶- از خطاها و اعمال بازگشتی جلوگیری کنید : در حال که پیشانی از انجام سریع یک کار مهم است ، بدون حفظ کردن عملیات

نیز مهم است . همانگونه که عامل در ارتباط با یاد به گونه ای طرا می شود که از سوئیچ کردن و سروریزی بازگشتی جلوگیری کند
به طور کلی ، تمام اصول لا پیش ذکر شده در این فواید می تواند به عنوان مورد نیاز است . این لا تسلیک های مهم ، عرضه کردن یا
دخواست کردن تنها اطلاعات مربوطه را کار تنها در زمان مورد نیاز است . این لا تسلیک های مهم ، عرضه کردن یا
برای این تسلیک هسته همچنین داشتن سیستم ، کار را نیازمند می کند که از این احتمالات انتخاب کرده عموما یک روش
این تر هست به های اعتماد دیگر یاد آوری .
بعد از انجام کردن اصول و تسلیک های قویتر داده شده در اینجا همیشه یک شناسی هست که که کار را شناختن
کند . بنابراین ، یک ویژگی را به دفع و در من حال را در این به تراوش می بردن ، اجازه دادن یک عمل بازگشتی است .
این عمل کاربر را به شرایط بهتری در اختیری تری می دهد .

۷- سادگی: سادگی بازی کردن به یک ویژگی که بازتاب دهنده عملیات های متفاوت در زندگی روزمره است. به عنوان مثال، این ممکن است گول زنده باشد که یک زندگی واقعی و حالت های ارتباط را بازی یک ارتباط با کامپیوتر متغیر با بزرگداشت کنیم. شاید یک راه حل بهینه این باشد که مدل ارتباط را ساده ای کنیم. همانی زندگی واقعی، استمراری ماهیت ادراکی و استمراری یک Task. یک ارتباط ساده یا ساده ای قابلیت حاصل کردن خواسته داشت، یک شی ای که در درک درونی و ذهنی ما ظاهر می شود که این باعث می شود مستعد یک شود به هویتی که ارتباط تقریباً نیازی به هیچ گونه ادراکی ندارد.

مثال ها: مورد ۱: فرضاً ما داریم یک بازی HyperCasual می سازیم. نوع طراحی آن باید بازی دره سنی ESR+25 باشد، تمام صفحه بندی، گرافیک، طراحی دکنده و صفحه چندصفحه ای افراد حرفه ای بازی است. بازی ما باید فرضاً یک عامل بازی ESR+3 داشته باشد. بازی ما باید یک برنامه حسابرسی، محیط آن باید به گونه ای باشد که یک فرد حسابدار که اطلاعات متوسل از نرم افزارها مختلف دارد بتواند به راحتی با آن کار کند.

مورد ۲: فرضاً در همان بازی، بازی مورد نظر ما از Action-Shooter باشد. ما اگر چه بازی را در همان محور پیش ببریم و از سبک خارج نشویم. با فرضاً برنامه حسابرسی، یک حسابدار انتظار دارد تقریباً تمام موارد مورد نیاز او در صفحه تعاملی در برش باشد و برای یافتن چیزی نیازی به جستجوی زیاد و پیچیده نباشد.

مورد ۳: در مورد بازی، خزنات حساسی یک کاراکتر نباید از یک حساسی بالاتر رود که کاربر نتواند برای جلوگیری بازی مشکل بخورد. فرضاً چند Item در گوشه های صفحه بندی از Task های بازرگانی مثل داریم تا پیشرفت خود در طول بازی را ببیند. در مورد حسابرسی: ما باید به این تفاوت که Item ها یا Widget ها که در منوی (مهرت) در دزدی میگذاریم مربوط به امور فاکتورها، مشتری ها، مبالغه هایی در طول ماه و...

مورد ۴: فرضاً منوهای جنبی های یک بازی (مانند Call of Duty) عموماً به یک صورت، بازنگری های کوچک در یک وقت از صفحه نمایان می شود. یا فرضاً در برنامه حسابرسی، بازنگری های فرله داده شده در صفحه های مختلف پروژ و سامانی (برنامه سیلندر) در سرهای خود مابقی است.

مورد ۵: فرماده بازی هنگامی که بازی را انجام دادیم کامل یا Task ای را به پایان رساندیم یا achievement ای گرفتیم، یک پیام pop-up شود

با فرماده برنامه بسیار پس از ثبت یک فاکتور، یک پیام پس از ثبت آن پیوسته شود که تاکنون فاکتور ثبت شده است

مورد ۶: فرماده بازی، آنرا از روند بازی طرایی کنیم که کاراکتر پس از هر نقطه ای از بازی، پیوسته بازگشت از قسمت بازی باعثی مرحله شود. بدین منظور Checkpoint هایی تعیین شود که پس از رفتن کاراکتر، بازی از آنجا ادامه پیدا کند
این برنامه حساسیتی بسیار، اثر فیزیکی اشتباه شد هنگام ورود اطلاعات، اگر اشتباهی رخ داد
توان به سادگی بازنویسی این برنامه و اندروید بازگرداند

مورد ۷: فرماده بازی: اغلب ویدیوهای پیرامون بازی در دسترس نیست و این باعث شده است که کاربر با دیدن آنها به سادگی بخشی از بازی را جلو برد (زفا بازی Tekken یا Mortal Kombat)
این برنامه حساسیتی، ویدیوهای رایگان برای افرادی که با برخی امکانات نسبت به نسخه پیرامون داده اند
تعیین شده که از آنها برای انجام کار خود استفاده کنند