Chapter6

فصل 6

If you share our conviction that customer involvement is a critical factor in delivering excellent software, you will ensure that the business analyst (BA) and project manager for your project will work hard to engage appropriate customer representatives from the outset. Success in software requirements, and hence in software development, depends on getting the voice of the user close to the ear of the developer. To find the voice of the user, take the following steps:

اگر شما هم باور ما مبني بر اين كه مشاركت مشتري مهم ترين فاكتور در ارائه يك نرم افزار عالي است ، شما اطمينان پیدا خواهید کرد که تحلیگر کسب و کار و مدیریت پروژه برای پروژه شما به سختی کار خواهند کرد تا از همان ابتدا نمایندگان مناسب مشتریان را جذب خواهند کرد . موفقیت در نیازمندی های نرم افزار ، و همچنین در توسعه نرم افزار ، وابسته به رساندن صدای کاربر به گوش توسعه دهنده است . برای پیدا کردن صدای کاربر مراحل زیر را انجام دهید :

■ Identify the different classes of users for your product.

■ Select and work with individuals who represent each user class and other stakeholder groups.

■ Agree on who the requirements decision makers are for your project.

* دسته بندی های مختلف کاربران را برای محصول خود تعیین کنید .
* انتخاب کنید و کار کنید با افرادی که نماینده هر دسته بندی از کاربران و سایر ذی نفعان هستند .
* درمورد نیازمندی های تصمیم گیرندکان پروژه خود توافق کنید

Customer involvement is the best way to avoid the expectation gap described in Chapter 2, “Requirements from the customer’s perspective,” a mismatch between the product that customers expect to receive and what developers build. It’s not enough simply to ask a few customers or their manager what they want once or twice and then start coding. If developers build exactly what customers initially request, they’ll probably have to build it again because customers often don’t know what they really need. In addition, the BAs might not be talking to the right people or asking the right questions

مشارکت مشتریان بهترین راه برای جلوگیری از مسئه شکاف انتظاری است که در فصل دو "نیازمندی ها از دید کابر" توضیح داده شد . عدم توافق بین محصولی است که کاربر انتظار دریافت آن را داشته تا محصولی که توسعه دهندگان ساخته اند . این کافی نیست که یک یا دوبار از مشتری یا مدیر آن ها سوال بپرسید که آن ها چه مخواهند و سپس شروع به کد نویس کنید . اگر توسعه دهندگان چیزی را که مشتریان در ابتدا درخواست کرده اند بسازند این امکان وجود د ارند که مجبور شوند دوبازه آن را بسازند زیرا مشتریان اغلب نمیدانند که آن ها به طور واقع به چه چیز نیاز دارند . به علاوه ممکن است تحلیلگران کسب و کار با افراد درستی صحبت نکنند و یا سوالات درستی نپرسند .

The features that users present as their “wants” don’t necessarily equate to the functionality they need to perform their tasks with the new product. To gain a more accurate view of user needs, the business analyst must collect a wide range of user input, analyze and clarify it, and specify just what needs to be built to let users do their jobs. The BA has the lead responsibility for recording the new system’s necessary capabilities and properties and for communicating that information to other stakeholders. This is an iterative process that takes time. If you don’t invest the time to achieve this shared understanding—this common vision of the intended product—the certain outcomes are rework, missed deadlines, cost overruns, and customer dissatisfaction.

ویژگی های که کابران به عنوان خواسته های خود ارائه می کنند لزوما با عملکردی که برای انجام وظایف خود با محصول جدید نیاز دارند برابری نمی کند .براي بدست آوردن ديد بسيار دقيقي از نيازهاي كاربر ، تحليلگر كسب و كار بايد رنج گسترده اي از ورودي هاي كاربران را جمع آوري ، تجزيه و تحليل و شفاف سازي كند و دقیقاً مشخص کند که چه چیزی باید ساخته شود تا به کاربران اجازه دهد کارهای خود را انجام دهند. اين يك پروسه زمان بر است كه به صورت تكرار شونده مي باشد . ا. اگر برای دستیابی به این درک مشترک - این چشم انداز مشترک از محصول مورد نظر - وقت صرف نکنید، نتایج معین عبارتند از دوباره کاری، ضرب الاجل های از دست رفته، بیش از حد هزینه ها، و نارضایتی مشتری.

User classes

كلاس هاي كاربري

People often talk about “the user” for a software system as though all users belong to a monolithic group with similar characteristics and needs. In reality, most products of any size appeal to a diversity of users with different expectations and goals. Rather than thinking of “the user” in singular, spend some time identifying the multiple user classes and their roles and privileges for your product.

مردم اغلب در مورد "کاربر" برای یک سیستم نرم افزاری صحبت می کنند به گونه ای که انگار همه کاربران متعلق به یک گروه یکپارچه با ویژگی ها و نیازهای مشابه هستند. در واقعيت ، اغلب محصولات در هر اندازه اي براي كاربران مختلف با انتظارات و اهداف متفاوت جذاب هستند . به جای اینکه به "کاربر" به صورت مفرد فکر کنید، مدتی را صرف شناسایی کلاس های کاربری متعدد و نقش ها و امتیازات آنها برای محصول خود کنید.

Classifying users

طبقه بندي كاربران

Chapter 2 described many of the types of stakeholders that a project might have. As shown in Figure 6-1, a user class is a subset of the product’s users, which is a subset of the product’s customers, which is a subset of its stakeholders. An individual can belong to multiple user classes. For example, an application’s administrator might also interact with it as an ordinary user at times. A product’s users might differ—among other ways—in the following respects, and you can group users into a number of distinct user classes based on these sorts of differences:

بخش 2 انواع بسياري از ذي نفعان را كه پروژه ميتواند داشته باشد را توصيف كرد . همانطور كه در شكل 6-1 نمايش داده شده است . يك كلاس كاربر زير مجموعه از كاربران محصول است كه آن نيز زير مجموعه اي از مشتريان محصول است كه آن نيز زيرمجموعه اي از ذي نفعان است . يك فرد مي تواند مطعلق به چندين كلاس كاربري باشد . براي مثال ، ، ممکن است ادمین برنامه نیز گاهی به عنوان یک کاربر معمولی با آن تعامل داشته باشد. ممکن است کاربران یک محصول - در میان راه های دیگر - در موارد زیر متفاوت باشند، و و می توانید کاربران را به تعدادی از کلاس های کاربری مجزا بر اساس این نوع تفاوت ها گروه بندی کنید:

■ Their access privilege or security levels (such as ordinary user, guest user, administrator)

■ The tasks they perform during their business operations

■ The features they use

■ The frequency with which they use the product

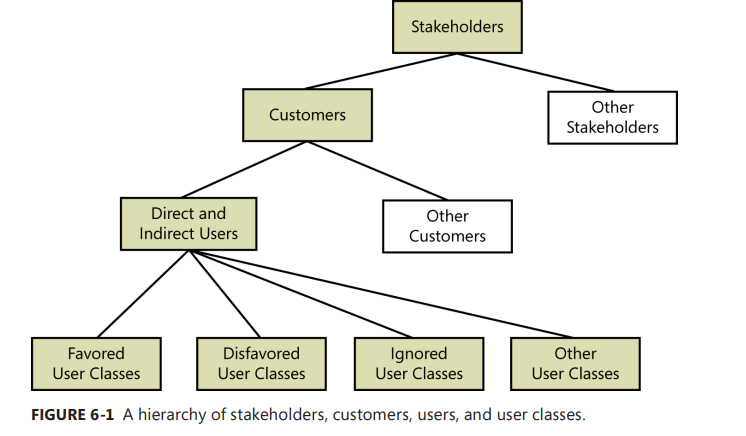
■ Their application domain experience and computer systems expertise

■ The platforms they will be using (desktop PCs, laptop PCs, tablets, smartphones, specialized Devices

■ Their native language

■ Whether they will interact with the system directly or indirectly

* رتبه بندي و سطح دسترسي امنيتي (به عنوان مثال كاربران معمولي ، كاربران مهمان ، ادمين ها)
* وظايفي كه آن ها در طول عملكرد هاي تجاري خود انجام مي دهند .
* امكاناتي كه آن ها استفاده مي كنند .
* تعداد دفعاتي كه آن ها از سيستم استفاده ميكنند .
* تجربه حوزه کاربرد و تخصص سیستم های کامپیوتری آنها
* سكويي كه آن ها به كار ميگيرند ( دسكتاپ ، لپ تاپ ، تبلت ، گوشی های هوشمند ، دستگاه ها ی خاص)
* زبان مادری آنها
* آیا ان ها با سیستم ارتباط مستقیم دارند یا غیر مستقیم



It’s tempting to group users into classes based on their geographical location or the kind of company they work in. One company that creates software used in the banking industry initially considered distinguishing users based on whether they worked in a large commercial bank, a small commercial bank, a savings and loan institution, or a credit union. These distinctions really represent different market segments, though, not different user classes.

گروه بندی کاربران به دسته بندی هایی بر اساس موقعیت جغرافیایی آن ها یا انواع کمپانی هایی که در آن کار میکنند بسیار وسوسه انگیز است . یکی از شرکت‌هایی که نرم‌افزارهای مورد استفاده در صنعت بانکداری را ایجاد می‌کند، ابتدا متمایز کردن کاربران را بر اساس اینکه در یک بانک تجاری بزرگ، یک بانک تجاری کوچک، یک موسسه پس‌انداز و وام، یا یک اتحادیه اعتباری کار می‌کنند، در نظر گرفت. این تمایزات واقعاً بخش‌های مختلف بازار را نشان می‌دهند، البته نه کلاس‌های کاربری متفاوت

A better way to identify user classes is to think about the tasks that various users will perform with the system. All of those types of financial institutions will have tellers, employees who process loan applications, business bankers, and so forth. The individuals who perform such activities—whether they are job titles or simply roles—will have similar functional needs for the system across all of the financial institutions. Tellers all have to do more or less the same things, business bankers do more or less the same things, and so on. More logical user class names for a banking system therefore might include teller, loan officer, business banker, and branch manager. You might discover additional user classes by thinking of possible use cases, user stories, and process flows and who might perform them.

یک راه بهتر برای شناسایی دسته بندی های کاربری فکر درمورد کارهایی که کاربران متفاوت با سیستم انجام خواهند داد . همه این نوع موسسات مالی دارای عابر بانک، کارمندانی که درخواست های وام را پردازش می کنند، بانکداران تجاری و غیره خواهند داشت. عابربانک ها همگی باید کم و بیش کارهای مشابهی انجام دهند، بانکداران تجاری کم و بیش کارهای مشابهی را انجام می دهند و غیره. . بنابراین، نام‌های منطقی‌تر کلاس کاربری برای یک سیستم بانکی ممکن است شامل عابر، مسئول وام، بانکدار تجاری و مدیر شعبه باشد. ممکن است با فکر کردن به موارد استفاده احتمالی، داستان‌های کاربر، و جریان‌های فرآیند و اینکه چه کسی ممکن است آنها را انجام دهد، کلاس‌های کاربر اضافی را کشف کنید.

Certain user classes could be more important than others for a specific project. Favored user classes are those whose satisfaction is most closely aligned with achieving the project’s business objectives. When resolving conflicts between requirements from different user classes or making priority decisions, favored user classes receive preferential treatment. This doesn’t mean that the customers who are paying for the system (who might not be users at all) or those who have the most political clout should necessarily be favored. It’s a matter of alignment with the business objectives.

اطمینان حاصل کنید دسته بندی های کاربران میتوانند مهمتر باشند از بقیه برای یک پروژه خاص.دسته بندی های کاربری مورد علاقه آنهایی هستند که رضایت آنها با دستیابی به اهداف تجاری پروژه بسیار نزدیک است . هنگام حل کردن تعارضات بین نیازمندی ها از دسته بندی های مختلف کاربری یا ایجاد اولیت های تصمیم گیری ، دسته بندی های مورد علاقه رفتار ترجیحی دریافت میکنند . این به این معنی نیست که مشتریان برای سیستم هزینه میکنند یا آنها که بیشترین نفوذ سیاسی را دارند لزوما باید مورد حمایت قرار بگیرند . این موضوع همسو با اهداف تجاری است .

Disfavored user classes are groups who aren’t supposed to use the product for legal, security, or safety reasons (Gause and Lawrence 1999). You might build in features to deliberately make it hard for disfavored users to do things they aren’t supposed to do. Examples include access security mechanisms, user privilege levels, antimalware features (for non-human users), and usage logging. Locking a user’s account after four unsuccessful login attempts protects against access by the disfavored user class of “user impersonators,” albeit at the risk of inconveniencing forgetful legitimate users. If my bank doesn’t recognize the computer I’m using, it sends me an email message with a one-time access code I have to enter before I can log on. This feature was implemented because of the disfavored user class of “people who might have stolen my banking information.”

دسته بندی های کاربری ناراضی گروه هایی هستند که به دلایل قانونی ایمنی یا امنیتی مجاز به استفاده از محصول نیستند . شما ممکن است به عمد ویژگی هایی رو ایجاد کنید که برای کاربران ناراضی انجام کارهایی که مجاز به انجام آن نیستند را سخت کند . برای مثال شامل مکانیزم های دسترسی امنیتی،سطح دسترسی کاربری، خصوصیات ضد بدافزاری و گزارش استفاده میشود . قفل کردن یک اکانت کاربری بعد از چهار تلاش ناموفق برای ورود ، از دسترسی دسته کاربران ناراضی (جاعلان هویت ) جلوگیری میکند ، البته با این ویژگی که باعث آزار کاربران واقعی فراموش کار می شوند . اگر بانک من کامپیوتری را که من استفاده میکنم تشخیص ندهد ، یک پیغام الکترونیکی با یک کد دسترسی یک بار مصرف برای من ارسال میکند که تا قبل از لاگ این کردن آن را وراد کنم . این ویژگی به وجود آمده به خاطر دسته کاربران ناراضی ازنوع افرادی که اطلاعات بانکی افراد را سرقت میکنند .

You might elect to ignore still other user classes. Yes, they will use the product, but you don’t specifically build it to suit them. If there are any other groups of users that are neither favored, disfavored, nor ignored, they are of equal importance in defining the product’s requirements

شما ممکنه که انتخاب کنید که دسته بندی های کاربری را نادیده بگیرید . بله، آن ها از محصول استفاده خواهند کرد ، اما به طور خاص شما آنرا مطابق با آنها نمیسازید. . اگر گروه دیگری از کاربران وجود داشته باشند که نه مورد علاقه، نه نادیده گرفته می شوند، در تعریف الزامات محصول از اهمیت یکسانی برخوردار هستند.

Each user class will have its own set of requirements for the tasks that members of the class must perform. There could be some overlap between the needs of different user classes. Tellers, business bankers, and loan officers all might have to check a bank customer’s account balance, for instance. Different user classes also could have different quality expectations, such as usability, that will drive user interface design choices. New or occasional users are concerned with how easy the system is to learn. Such users like menus, graphical user interfaces, uncluttered screen displays, wizards, and help screens. As users gain experience with the system, they become more interested in efficiency. They now value keyboard shortcuts, customization options, toolbars, and scripting facilities

هر دسته بندي كاربري مجموعه نياز مندي هاي خودش را براي كارهايي كه اعضاي آن دسته بندي بايد انجام دهند خواهد داشت . ممكن است همپوشانی بینی نیاز های کلاس های کاربری متفاوت ایجاد شود .

عابر بانک ها ، بانک داران تجاری، و مسئولین وام همه ممکن است مجودی حساب بانکی یک مشتری را بررسی کنند . همچمنین کلاس های کاربری متفاوت ممکن است انتظارات کیفی متفاوتی داشته باشند ، به عنوان مثال کارایی، که انتخاب طراحی رابط کاربری را کنترل خواهد کرد . کاربران جدید یا گاه به گاه نگران این هستند که یادگیری سیستم چقدر آسان است.بعضی کاربران منوها، رابط های کاربری گرافیک،نمایش های نامرتب ، جادوها و کمک های صفحه ای را دوست دارند . به عنوان مثال کاربرانی که در رابطه با سیستم تجربه کسب میکند علاقه بیشتری به کارایی می یابند . . آنها اکنون برای میانبرهای صفحه کلید، گزینه های سفارشی سازی، نوار ابزار و امکانات برنامه نویسی ارزش قائل هستند

Don’t overlook indirect user classes. They won’t use your application themselves, instead accessing its data or services through other applications or through reports. Your customer once removed is still your customer.

کلاس های کاربر غیر مستقیم را نادیده نگیرید. آنها خودشان از برنامه شما استفاده نخواهند کرد، در عوض به داده ها یا خدمات آن از طریق برنامه های کاربردی دیگر یا از طریق گزارش ها دسترسی خواهند داشت. مشتری شما پس از حذف همچنان مشتری شماست.

User classes need not be human beings. They could be software agents performing a service on behalf of a human user, such as bots. Software agents can scan networks for information about goods and services, assemble custom news feeds, process your incoming email, monitor physical systems and networks for problems or intrusions, or perform data mining. Internet agents that probe websites for vulnerabilities or to generate spam are a type of disfavored non-human user class. If you identify these sorts of disfavored user classes, you might specify certain requirements not to meet their needs but rather to thwart them. For instance, website tools such as CAPTCHA that validate whether a user is a human being attempt to block such disruptive access by “users” you want to keep out.

نیازی نیست که کلاس های کاربری حتما انسان باشند . آن ها ممکن عامل های نرم افزاری باشند که یک سرویس را از طرف یک کاربر انسانی انجام میدهند به عنوان مثال ، بات ها . عامل های نرم ممکن است شبکه را برای اطلاعات درمورد کلاها و خدمات اسکن کنند ، فیدهای خبری سفارشی جمع آوری کنند ، امیل های دریفاتی شما را پردازش کنند یا دیتا ماینیگ انجام دهند . عامل های اینترنتی که وبسایت ها را برای آسیب پذیری ها یا ایجاد هرزنامه جستجو می کنند ، یک نمونه از رابطهای کاربری غیر انسانی هستند . اگر این دسته از کلاس‌های کاربری ناراضی را شناسایی کنید، ممکن است الزامات خاصی را نه برای رفع نیازهای آنها، بلکه برای خنثی کردن آنها مشخص کنید. برای مثال، ابزارهای وب‌سایتی مانند CAPTCHA که انسان بودن کاربر را تأیید می‌کنند، سعی می‌کنند چنین دسترسی مخربی را توسط «کاربرانی» که می‌خواهید از آن جلوگیری کنید، مسدود کنند.

Remember, users are a subset of customers, which are a subset of stakeholders. You’ll need to consider a much broader range of potential sources of requirements than just direct and indirect user classes. For instance, even though the development team members aren’t end users of the system they’re building, you need their input on internal quality attributes such as efficiency, modifiability, portability, and reusability, as described in Chapter 14, “Beyond functionality.” One company found that every installation of their product was an expensive nightmare until they introduced an “installer” user class so they could focus on requirements such as the development of a customization architecture for their product. Look well beyond the obvious end users when you’re trying to identify stakeholders whose requirements input is necessary.

بیاد بیاورید ، کاربران زیر مجموعه ای از مشریان هستند ، که خود نیز زیر مجموعه از ذی نفعان هستند . شما نیاز خواهید داشت که طیف بسیار وسیع تری از منابع بالقوه از نیازمندی ها به جز کلاس های کاربری مستقیم و غیر مستقیم را در نظر بگیرید . . به عنوان مثال، حتی اگر اعضای تیم توسعه، کاربران نهایی سیستمی که می‌سازند نیستند، شما به نظرات آنها در مورد ویژگی‌های کیفیت داخلی مانند کارایی، قابلیت تغییر، قابلیت حمل و استفاده مجدد نیاز دارید، همانطور که در فصل 14 توضیح داده شد، "فراتر از عملکرد". " یک شرکت متوجه شد که هر نصب محصول آنها یک کابوس گران است تا زمانی که یک کلاس کاربری "نصب کننده" را معرفی کردند تا بتوانند بر الزاماتی مانند توسعه معماری سفارشی سازی برای محصول خود تمرکز کنند.

. زمانی که در تلاش برای شناسایی ذینفعانی هستید که ورودی مورد نیاز آنها ضروری است، فراتر از کاربران نهایی آشکار نگاه کنید.

Identifying your user classes

شناسایی کلاس های کاربری شما

Identify and characterize the different user classes for your product early in the project so you can elicit requirements from representatives of each important class. A useful technique for this is a collaboration pattern developed by Ellen Gottesdiener called “expand then contract” (Gottesdiener 2002). Start by asking the project sponsor who he expects to use the system. Then brainstorm as many user classes as you can think of. Don’t get nervous if there are dozens at this stage; you’ll condense and categorize them later. It’s important not to overlook a user class, which can lead to problems later when someone complains that the delivered solution doesn’t meet her needs. Next, look for groups with similar needs that you can either combine or treat as a major user class with several subclasses. Try to pare the list down to about 15 or fewer distinct user classes.

در اوایل پروژه کلاس های کاربری مختلف را شناسایی کنید تا شما بتوانید نیازمنیدها را از نمایندگان هر کلاس مهم استخراج کنید . یک تکنیک کاربردی برای این یک الگوی همکاری است که توسط الن توسعه پیدا کرده که "گسرش و انقباض" نامیده شده است . با سوال از اسپانسر پروژه شروع میشود که چه انتظاری از سیستم دارد . سپس هر تعداد از کلاس های کاربری که فکرش را می کنید طوفان فکری کنید