The “3 E’s” of Educational Software Design: Education, Engagement, and Ease of Use

Lynn Rosener

Summary

Providing appropriate educational content in an engaging game that is easy to use is the key to a successful educational product. This paper covers the “3 E’s” — the three categories of design principles relevant to educational software: educational, engagement, and ease of use.

Most of the design principles presented below are appropriate for children ages 10 and under although many are also appropriate for older children and even adults. These principles can be used as a guide during the design phase as well as used as a checklist after initial designs have been specified. However, for these principles to be truly useful during design, specific guidelines for each of these principles must be specified. The specific guidelines will vary depending on the academic focus of the software product as well as the age of the intended users. (In this paper, a few guidelines have been included when necessary to clarify a specific principle and in no way represent comprehensive coverage of that principle.)

This report concludes with an introduction to usability testing — to emphasize the importance of iterative design and to remind designers to stay close to the users of our products, the children.

Educational Design Principles

1. **Make the learning content important and central to game design.**
   Start your design with a solid understanding of the educational content you will be presenting — and make the educational material integral and central to your design. Don’t trivialize the educational content by “infusing your game with educational content” at the end of the design process. Avoid falling into the trap of creating activities where educational information is presented to the child only in “drill-and-practice” or multiple-choice sessions. Instead, challenge the child, in the game, to solve problems in a logical situational context -- using the knowledge you’ve specified in the educational goals. Give children what they need to know, when they need to know it. Then, give the child chances to practice and apply what they have learned.

   When the educational material is central to the design, players will be able to develop a sense of mastery over the educational content as they play the game — and developing a sense of mastery is a sweet reward. So there’s no need to sugar-coat the learning in a product (through technical “bells and whistles”) if it’s presented properly. Extrinsic rewards need only be provided, as necessary, to support the other aspects of game play such as engagement.

2. **Provide real-life, meaningful, educational experiences.**
   This doesn't necessarily mean that all things have to be "true-to-life". In fact, providing a fantasy environment in which real-life experiences happen is sometimes even more effective because it taps into the child's imagination. But it does mean that children should be presented with real problems to solve or placed in real situations for learning.

   One way to do this is to present the familiar in a new or different way so that the child can freely explore the educational content in an engaging and, yet still, meaningful way. The real intent of this principle is to provide children with exposure to the kinds of problems they will have to face in the real world and with situations that allow them to learn and practice solving those problems using the specified educational content in a "safe" and engaging way.

   Set clear challenges and/or clear objectives for the children to accomplish. Requiring the children to respond to questions with multiple-choice answers doesn't elicit higher-level thinking nor is it very engaging.

3. **Have the child interact with the educational content in a way that provides a sense of "wholeness" rather than "dis-jointedness".**
   We want to ensure that the educational content of our games is correct and true to the curriculum, but how we get the child to that information is the key. While some knowledge can be broken up into small pieces and fed to children so that they can absorb and reap the "educationally nutritious" benefits, this approach often drains the educational content of what feels real and meaningful to a child — and makes it boring as well.

   Try to present the educational content as a coherent whole. This task is not as easy as it might sound. In support of this, make sure you can complete the sentence: "By knowing the educational content in this game the player will gain a better understanding of or gain more mastery in ...". Use the answer that you give as a guide to whether or not you have achieved your educational goal.

4. **Make sure you design affective, as well as cognitive, content into the game.**
   Affective means "influencing feeling or emotions" and children's learning capabilities are strong when linked to affective content, especially during the early years. By providing affective meaning to our educational content, we can help children discover a richer understanding of the cognitive content and do so in a more engaging way. This is also true for curriculum areas such as math and science where, unfortunately, the affective associations have been totally lost. We can use gameplay to bring these affective components back into meaningful learning.

   Note: There is a third domain education is concerned about: psychomotor—having to do with physical movement. We should look for opportunities to build this in as well.

5. **Drill-and-practice activities can be used in specific situations, but use them sparingly and appropriately.**
   If in your game, there is a place where the child has to increase mastery over some skill to achieve one of the game's goals, then a drill-and-practice activity may be appropriate.

When used appropriately, this type of activity, which is usually perceived as boring and stressful, may actually add to your game play excitement. But be clear: this should not be your primary approach, unless mastery over a specific skill is your educational objective. Use this type of approach sparingly. Educators are moving away from rote learning and this type of approach.

6. **Try to structure learning experiences that have the child do something that can be done best with a computer.**
   Look for learning experiences that can't be done any other way or that the computer can provide in a unique way. Some examples include: manipulating three-dimensional shapes in a geometry lesson; selecting background music to create a mood for a child's multi-media presentation of a written report ; or retelling a story by changing the perspective of who's telling the story.

7. **Create situations where cooperative and collaborate learning can be used.**
   Educational reform is moving more toward children working in groups and the social aspects of learning. Take advantage of this in your game design. Create possibilities for group play or multi-player interaction. As telecommunications capabilities become more common in the home and school, this will become a quickly growing area in education software.

Engagement Design Principles

1. **Be able to explain what the game's objective is and how to play in "ten words or less".**
   Obviously while "ten words or less" is not a hard and fast rule, the intent is rather to make sure that the overall, driving goal of the game is easy to understand and can be communicated from one child to the next. Having a clear, simple objective for the game, stated early in the design process, will also help you keep focus and make specific design decisions as they arise.

2. **Create a robust, rich world for the player to interact with and explore.**
   Tap into the child's imagination and create a space for them that is full of the wonder and joy that learning is all about. Make it rich enough that they can construct this new world in their head and that it becomes real, but be careful about over-designing it. Create spaces where a child's imagination can fill the void and add depth in a way meaningful to that child.

3. **Focus on the game and its educational and engagement value first—make sure you get it right before you add the sizzle.**
   There's a tendency to be enticed by the newest and greatest technology sizzle and then design a game around it. Instead, design your game around your educational and engagement objectives and then use what technology has to offer to support those objectives.

4. **Don't spare the sizzle.**
   After creating an educationally sound and engaging game, let the animators and sound people loose. It's a mistake to think that because we do children's games, these games don't require all the sizzle that technology has to offer.

5. **Create a comfortable gamespace for players.**
   The less unknown and/or new information presented in the game and the fewer mental models players have to develop, the easier children will be able to grasp and play the game. The corollary of this statement is that the more similar the game is to something the players already know, the more comfortable it will feel. Ways of doing this include using recognizable characters; using well-known storylines or environments, familiar game formats, objectives, and strategies, etc.

6. **Design for simple rules, but complex interactions.**
   This doesn't mean that the game has to be simple or easy. What it does mean is that the complexity of the game comes from the educational content and the problem-solving activities. The how-to aspect of the game should be simple, clear, and easy for the player to remember.

7. **Design the interface so that it is transparent to the player.**
   The interface should not intrude or get in the way of the gameplay and should be innovative only if it enhances play. Instead of putting your creative juices into making the interface bold and different, put your creativity into designing a way to get the educational content across in an engaging way.

8. **Provide an opening sequence that "grabs," but have it last "just long enough" to arouse emotions and not so long that it irritates players.**
   Players want to get into and out of the game immediately. While amusing the first or second time, these opening sequences become just another mouse click required to get into the game and really add little to the game. If you need to use opening sequences to define your storyline and provide context to the game, rethink your design.

   This is especially true for CD-based games where the temptation to use additional graphics, sound effects, and music is strong. When contextual material needs to be presented initially to players, build it into the gamespace as a place or activity. Start players from these places, but design them so that the player can avoid them later on. In fact, you may be surprised to discover that players want to return to this contextual material at a later time after playing the game a little. However, designing the game so that they have to restart the game to see it—or to have to view it every time they start the game—is not desirable nor optimal.

9. **Provide an easy entrance into the game. Make it accessible to first-time players.**
   We want children to play our games for a purpose: to enhance their knowledge. So don't create games where players have to do "unnatural" or extremely complex things when first playing. Make sure that playing the game is fair (necessary information needed to solve the game's puzzles is not excessively difficult to discover) and allows everyone at all levels of gaming experience to enjoy it.

Present the gameplay in a way such that it can be played on a simple level initially and that players can work into the complexity over time. For example, a good teacher makes a difficult concept accessible by simplifying first and then building more complex models upon the easier ones. The consequence of not designing this way is that gameplay that is so complex that many players never get to the core educational value of the game.

10. **Reward players early and often.** ***Corollary:*** **Don't clobber players at the beginning with little or no reward, or worse yet, punish them by making them redo actions they've already performed.**
Instant gratification isn't fast enough because players are people and people feel stupid when they can't figure things out. Don't exacerbate the negative feelings players are already generating by themselves as they are learning. Make the feedback appropriate. Obviously, we don't want to go overboard on simple successes like "1+1=2 Fantastic!". It won't ring true to the players. Positively reinforce positive behavior and provide guidance when giving negative feedback.

    On a similar vein, making players start over from the beginning (i.e., ground or points gained are lost) whenever they make a mistake (or several mistakes) is "old-think" in game design and will result in a strong disincentive to even the youngest players. I am not referring here to situations where children make a mistake and get to try again and again, nor am I referring to situations where the child has obtained some object which is lost due to natural game play, but rather when the "go back to the beginning and start again" response is implemented merely to make game play more difficult.

11. **Give people clues when they're stuck and make the hand holding as natural as possible and/or part of the game design.**
Educational games should not be tests of machismo. They're supposed to be designed so the children can have fun while learning something. So help them out in ways that don't impose on the engagement or learning.

12. **Create real characters that players can relate to and recognize. Represent the diversity of your players and avoid discrimination and bias.**
Make sure you give your characters a face so players can feel like they're working with or against real partners or opponents and not just the computer. People are wired to respond favorably to faces, especially those with proportions of young children (larger eyes relative to head size, etc.). Keep in mind that although the computer can be used as a catalyst for social interaction, people feel better when they are perceiving that they're interacting with something living. Before you create a character, do a complete character description. Know the character inside and out—warts and all! Make the characters believable—not too perfect.

    Be sensitive, however, to the diversity of players. Be sure to be as inclusive as possible and avoid stereotyping and other forms of discrimination and bias.

13. **Use extrinsic rewards and scoring appropriately.**
Whenever possible, use extrinsic rewards and scoring only as redundant reinforcement (not necessary) to the learning process. If additional rewards are needed for game play, make sure they are used appropriately for your target players. Recent data suggests that

boys and girls may have different attitudes about the type of rewards and scoring mechanisms they prefer.

14. **Use "nuisance characters" appropriately.**
    While "nuisance characters" (those pesky little characters that appear and impede your progress or steal your stuff) may add excitement and challenge to a game, be aware that not all players respond positively to this type of game play. You may want to provide this aspect of game play as an optional feature.

15. **Look to the "oldies, but goodies" games for ideas about what makes them fun and addictive.**
    Look at board games as well as early computer games. What we assume on a first glance to be engaging, may on a deeper look be something quite different. When you hear children or adults discussing why they like a particular game, ask them. Make your life's passion to discover what makes games (and specifically computer games) fun and engaging to play and incorporate that knowledge into your game design!

16. **Violence is not okay in any form.**
    While some designers have made the distinction between aggression (where no one gets hurt) and violence, avoid placing the child in any situation where they must encounter violence or use violence to play the game (unless the game is specifically directed at helping children cope with violence in their lives). Violence should not be used in any form of entertainment because rather than being cathartic, it has been shown to lead to increased violent behavior in children.

Ease of Use Design Principles

1. **Know your players and design for their specific interests and capabilities.**
   Make your assumptions about players' educational and gaming capabilities explicit. Don't assume that you or your own child is a typical player. You are not. You have more contact with computer hardware and software—and especially games—than average players. Although we develop software for a wide variety of players and no "typical" player can be specified that covers all design concerns, it is important to think clearly about who will be using your software and to make your assumptions about players explicit.

   The reasons for this principle are obvious, yet surprisingly it is often overlooked or merely given lip service. In fact, it is key to developing games that sell well.
   - Develop a player profile for each different type of player you expect will use your game with enough developmental detail to guide game design.
   - Create typical scenarios (descriptions of player activity flows) for each of the player profiles.
   - Use this information during the development of your product.

2. **Make sure players can develop a consistent mental model of the game.**
   Design the game so that the player can easily develop a mental or conceptual model of your game overall. Once a player develops a model of part of the game, he or she extrapolates about how the rest of the game will work. If a game design is consistent, the player's predictions will work more often than not and the game will appear easier to learn and use.

   Even a few minor exceptions to the rule will force the player to create a more complex model, making the game appear more difficult to learn and use. In addition, players become especially aware of inconsistencies when context switching across activities or games is possible. Therefore, inconsistencies should be avoided whenever possible.

   This is an area where many well-intentioned game designers often have trouble — making aspects of their game consistent which don't need to be and other which should be consistent, are not. Unfortunately, much of the determination of when to be consistent and when not to be consistent involves judgment and experience and is difficult to specify clearly with simple rules.

3. **Make game design as simple as possible.**
   Simplicity is achieved through good game design by adhering closely to the player's conceptual model of the game. Simplicity in design can be achieved through screen design as well as functionality.

   Simplicity in Screen Design

   - Make sure that separate areas in a screen are visually distinct and clear to the player.

   - Don't pack the screen with information. While you want to avoid forcing the player to go to multiple screens for a given activity, it is often more effective for the player if you take a complex, dense screen and split it into two, more usable screens.

   - Try to control the number of choices or options you present to the player at any given time. If you have to present a long list of options, use the "chunking" technique which supports short-term memory capability. "Chunking" refers to clustering things into groups of five to seven items which people can work with more easily. Five to seven clusters of five to seven items is much more reasonable than a list of 25–49 items on a screen. For this technique to work, the items grouped together do not even have to have anything in common, although if that is possible, it helps.

   - Lists of options should be listed vertically rather than horizontally for easy scanning.

   - Use color appropriately. Certain cultural groups attribute very specific meanings to certain colors. Using these colors inappropriately can cause problems. (An example would be to use "red" in the USA for a "go-ahead" function instead of the more expected "green".) In addition, people are neurologically wired to associate

things of a similar color with other similarly colored items. Make your color choices carefully and not just according to aesthetics.

Simplicity in Functionality

Simplicity of design does not necessarily mean a limited set of features nor does it exclude complex capability; rather it refers to the quality a player perceives when a thoughtful, consistent conceptual model is presented.

- As much as possible, use a structure and terminology that matches what most of the players will bring to the activity domain.
- Avoid mode or state changes. They require the players to adjust to the system rather than stay focused on their activities.
- Provide lists for the player. It is easier for the player to recognize information on the computer screen rather than recall it.

4. **Provide players with information to orient themselves in the gamespace.**
Often players learn to play a game through exposure to a small subset of the game's total design. These players may have a difficult time, when they make a mistake or get lost, if they are not aware of the game's overall structure and where they are relative to that structure.
   - Design the game so that its overall structure is as clear and obvious to the player as possible.
   - When appropriate, provide the player with cues about where he or she is, what he or she is doing, what he or she has as tools/prizes, and what he or she can do next.

5. **Provide players with appropriate feedback.**
People build mental models that may or may not be valid. Feedback is the mechanism we use to continually adjust our models, and in effect, perform a reality check. Without appropriate feedback, people may continue to work with a mental model that is invalid. In addition, many people feel uncomfortable and anxious without enough feedback. So it is important to provide appropriate and adequate feedback for every action that the players make.

   Software user interfaces are, in fact, a dialog between the player and the computer. Look to basic rules of effective communication as a guide. Just as it is frustrating to talk to someone who just grunts, or worse yet, does not respond at all, it is similarly frustrating for players to interact with a game that does not provide adequate feedback. Game design must provide feedback of some form for every action that the player makes. It may be communicated through text, sound, or graphics, but it should always be immediate and appropriate for the situation.

6. **Empower players so that they are in control of the game at all times.**
Let players control all appropriate aspects of the game; be cautious about doing things automatically "for" them. Mastery of the environment is a basic need of all people. Our players use our products to facilitate learning and to have a good time; they do not want the computer to "take control" or to direct them. Players need to feel in control of the relevancy, amount, and speed of the information being presented to them. This is true to some extent even for the youngest players.

- Don't have the game perform actions for the player without the player's consent or explicit direction unless you feel fairly comfortable that the player is expecting just that action. For example, don't move the player to a logical spot in the game unless he has specified it.

- Present only information that is relevant to the activity at hand. Let players access other information when they need it.

- Let players control the start, stop, and rate of information presentation. These principles are necessary, not only because different players process information at varying rates, but because an individual player's ability to process information changes as he learns and develops proficiency with the game. For example, when text is presented, do not time it before removing it. Always allow the players indicate when they are done. and ready to move on.

Just as players want to control their environments (game), so do software designers. Avoid designing games that control the environment at the expense of the player's control. For example, avoid having the game prompt the player for information out of sequence from the player's natural process. While it may be that it is more efficient for the software to capture the information at an earlier time, if the result is that the player feels that the game has taken control and is driving the player, the interface has not been well designed.

7. **Provide flexibility in game to accommodate players needs.**
Design the game so that 80% of the actions and responses possible are inherent in the base (normal) gameplay. Provide for the remaining 20% by providing customization and other types of options. Specifically, flexibility in game design refers to:
   - Providing an interface that can accommodate a variety of player needs and skills.
   - Accommodating a variety of player errors.
   - Designing a "forgiving" interface, e.g., allowing the player to back up a step or two, rather than having to restart.
   - Trying to design a simple set of operations that provide good coverage, yet may not handle every conceivable action.

   While creating flexibility in a game provides players with some latitude in accomplishing their tasks, it does not mean that players should be provided with several ways of doing any given task. In fact, providing different ways of doing the same thing actually creates greater "conceptual load" for players. It makes the mental model players create more complex and leads to greater player errors and frustration.

8. **Provide players with a successful experience every time.**
Design the game to maximize a successful learning and gaming experience. In the ideal case, a game would need no error messages. It would be structured so that the player can only try to do legal things. Approaching this goal requires careful design of the overall game. But attempting to follow it can help the designer as well as the player. Error checking usually takes lots of code; changing to a better design can be free. When errors do occur, try to design built-in recovery for the player.

9. **Provide players with an experience which builds self-esteem.**
   Design your game so that it makes players feel good. People like to feel good about themselves and they will be more apt to purchase and use games that build up their self-esteem. This does not necessarily just mean providing messages like "Good Job". It means that you also need to think about designing the software so that you help the players cope with and play the game successfully.
   - Design so players can learn and experiment with game relatively risk-free.
   - Design the game so the player can accomplish minimal tasks successfully from the beginning.
   - Protect the player from disastrous actions.
   - Embedded training and help should be an integral part of the external design—not left as "last minute" design activities.
   - Use positive, constructive messages. Avoid negative and condescending language.
   - Avoid unnecessary computer jargon or concepts that may confuse or intimidate the players.

10. **Design Iteratively by Testing with Representative Players**
    Plan to design iteratively by testing with representative players to insure these principles are in force. No one is such a great designer that they can anticipate all the problems players will have. Let your players tell you how where they are having problems, and usually, they will tell you (or at least indicate) how to solve those problems too! Conducting usability tests during development is one of the best ways to obtain this information from your players. The following section presents an overview of usability testing.

Usability Testing to Support Iterative Design

What is Usability Testing?

Usability testing is a process used to evaluate users' ability to use and their satisfaction with a given product. Within the context of educational games, the goal of usability testing is to evaluate the usage performance (not learning performance) of children interacting with the game and the engagement and/or satisfaction they experience while using it.

The term "usability testing" actually encompasses a large range of testing methodologies. Although to some, the term "testing" implies evaluation processes with stringent controls (resulting in outcomes which are statistically significant), "usability testing" is commonly used to refer to an observational form of evaluation using relatively small numbers of users (3-5 per age group). The significant aspect of this type of testing is that it uses representative users performing representative tasks or activities to uncover major flaws in the design.

Why Conduct Usability Testing?

As the market becomes more saturated with products claiming to be "educational" and "easy to use", consumers will become more able to discern which products actually deliver on their

promises. Usability testing is one way to keep up with this growing market and to support the development of successful products.

Identifies Critical Usability Problems

Although it is not possible to produce results which *validate* the ease of use or engagement of a product's design, usability testing is conducted with sufficient controls to be able to *identify critical defects* in the product's user interface. (The term "user interface" refers to everything that the user interacts with when using the product: from the screen elements and the interaction with them; to the input devices and other related hardware; to the documentation and any training materials. Regardless of how developers separate these aspects of the product, the user does not, so for the purposes of testing, they all must be considered.) Identifying critical user interface defects is the single most important outcome of this type of evaluation.

Usability is related to the effectiveness and efficiency of the user interface and to the user's reaction to that interface. Usability testing focuses on four basic usability attributes: user productivity, ease of learning, user satisfaction and engagement, and product effectiveness. Within the context of these four attributes, the nature of the problems usability testing uncovers are:

- **User Productivity** — problems users experience when using the product for their intended purposes. This attribute is often referred to as "ease of use". User retention and the ability to use the product over time could also be included here.
- **Ease of Learning** — problems users encounter when learning to use the product.
- **User Satisfaction and Engagement** — areas in the user interface which cause dissatisfaction or are an obstacle to an engaging experience.
- **Product Effectiveness** — problems with the product delivering what it was intended to provide for the user - educational content and gameplay and engagement.

Usability Testing Activities

Usability testing can be conducted at any time during product development. However, it is most effective when conducted early in the development process when it is much easier (and less costly) to make changes to the design. When usability testing is conducted early on, subsequent testing will typically indicate refinement needed in the design - changes that are relatively easy and inexpensive to make. On the other hand, if usability testing is not initially conducted until later in development, major design flaws are often observed which indicate changes that are relatively costly to repair. Often, project reality at this point in development is that the changes don't get made because there isn't the time or resources available.

In fact, to be most effective, usability testing should be followed by product design revision and retesting -- a process commonly called "iterative design". The bottom line is: don't conduct usability testing unless you plan on iterative design and are committed to making the necessary changes identified during the testing process.

The Key to Effective Testing and Design

Crucial to conducting effective testing and designing successful products is an understanding of our users and why they use our products. The two critical areas to focus effort are:

- Understanding the users, children -- their developmental and learning capabilities, as well as how they interact with relevant technology.
- Understanding how to design educational content in increasingly engaging ways.

Testing Activities in the Product Life Cycle

For each life cycle phase, the most appropriate testing approach is described. However, since user interface design (and development) doesn't always happen in the linear fashion presented below, different interface modules may require different testing approaches since some modules may still be in the concept stage while others may already be in early code development. This is meant only as a guide.

Product Investigation - Research and document child development, learning theory, and educational/curriculum areas. While not a testing activity, a child/user profile (reviewed with marketing) and educational goals should be developed at this time -- to drive the design of all the testing activities that follow.

Design Specifications- Either low- or high-fidelity prototype testing is possible and appropriate at this time. Low-fidelity prototyping refers to what's commonly called "paper-and-pencil" prototypes -- made with common office/art materials. High-fidelity prototyping refers to the use of code or authoring/development tools such as MacroMind Director or HyperCard. If a revision to a current product is being developed, then the old product may be used at this time, mixed with low-fidelity supports where the changes are proposed.

In fact, it is often customary at this point in the development cycle to test several, alternative designs rather than just one. **This is the time to conduct the first usability test** to really make sure that your approach is correct before too much is committed to code.

Development - As the design is implemented in code, free-standing, logical modules of the product may be tested on-line. This is the best time to test for interaction (input/output responses) and user performance issues as well as concerns with specific user interface design features (graphics, navigational concerns, etc.).

Pre-Alpha - This will be the first time that the product can be evaluated as a whole. This is a critical time for the product. If all modules have been tested and major flaws have been repaired, then the major focus of an evaluation at this time is to refine the overall gameplay and/or global navigational design issues. **This is not the time to conduct the first usability test.**

Alpha - This is really the time for acquiring data on longevity use and play issues which can only be captured with extended use. However, time will be short to make any significant changes. Critical user interface problems encountered at this phase will

require a decision to either delay shipment or ignore the data obtained -- both serious consequences.

Post-Release - Data on usage problems can be collected at this stage, not only for revisions to current products, but also to better our understanding of our users and how to better provide for their needs .

Comparison of Usability and Quality Assurance (QA) Testing

While usability testing and QA testing share some common elements, they actually differ in two significant ways. To help avoid any potential misunderstandings, the similarities and differences are listed briefly below.

Similarities

- The goal in both activities is to improve the product.
- The activities locate problems not recognized by the development team.
- Neither of the activities requires a finished product for effective evaluation.

Differences

- The focus of the usability testing is on the *behavior of the user* rather than on the *product's features.* While product features and functions are evaluated, they are only evaluated in light of how they are received by the end user. With usability testing, it is unusual for all features to be tested, as opposed to QA activities where overall consistency and correctness of feature performance is tested.

- Testers used in usability testing are representative users of the product, in our case, children. They can be expected to make the kinds of assumptions about how the product works that any other of our prospective users would make. QA testers, on the other hand, are not representative users -- and the assumptions they make about how the product should work are defined by the designers.