

Guidelines for User-Centered Design

by [Kentaro Toyama](#), *Principal Researcher, Microsoft Technology for Emerging Markets Group*

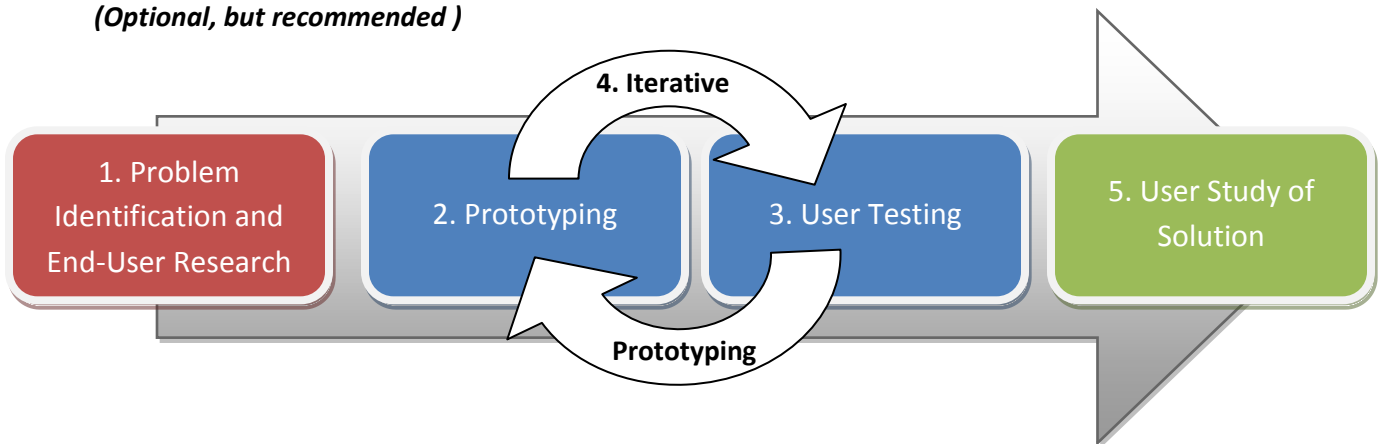
User-Centered Design (UCD) is a broad approach to technology design, in which the constraints and desires of the eventual user of the technology are taken into account throughout the design process.

It might seem obvious that user needs should be taken into account, but in practice, many engineers and software developers end up focusing so much on technology and features, that they forget to incorporate actual users into the invention and development process.

Although there are many schools of thought about how to do good UCD, most recommend steps like the following (click on each step to learn more):

Steps for Conducting User-Centered Design

- 1) **Spend time with actual users or potential end-users to identify challenges they face, often with respect to a particular issue.**
- 2) **Prototype potential solutions.**
- 3) **User-test to see how the prototypes work or don't work.**
- 4) **Iteratively prototype and test, repeating steps 2 and 3.**
- 5) **Conduct a rigorous user study of your best solution.**
(Optional, but recommended)



Also be sure to check out some [Examples of User-Centered technologies](#) from Microsoft Research India's Technology for Emerging Market's Group.

1) Spend time with actual users or potential users to identify challenges they face, often with respect to a particular issue.

- **Identify a problem domain and a community to work with.** In some cases, you may have a good idea of both (*e.g.*, math education among the young kids in a particular city slum). In other cases, you might have either a particular community in mind, or a (*e.g.*, a rural village near your home, or a healthcare non-profit) or a rough idea of what kind of impact you'd like to have (*e.g.*, better care for tuberculosis patients).
- **Find an organization that works in the community (or *is* the community) you hope to impact and on the kinds of issues that you want to impact.** Many organizations will already have built trust with potential users with whom you want to interact and you should see if any of them are interested in working with you on your project. Non-profit, non-governmental organizations are often good partners, so one resource you might consider using resources like friends, family, and the [TakingITGlobal Organizations Database](#) to find one near you.

You may have to knock on several doors before you find a good fit, but ideally, you can find an organization works in a place that is easy for you to reach so that you can visit often. Remember, even the wealthiest countries have communities in need of socio-economic development so you shouldn't have to travel far to find communities that could benefit from your ingenuity.

- **Spend as much time and with as many people as you can, who might be impacted by your project.** Talk to them, observe them as they go to work or meet with friends, even live with them (if you can!). Get a sense for the challenges that they face with respect to the area you're interested in, and learn as much as you can about existing solutions, known challenges, etc. The more you know, the more it will help you later.

Some hints during this process...

- Establish rapport first – you may have to engage with people on the things they want to talk about initially. Then, as they warm to you, you can ask more questions that you're interested in.
- Take a lot of photos – photos are great for a variety of reasons, including helping you to remember what you observed, helping you to brainstorm, and providing good visuals during a presentation (But, ask for permission before you do; some people may be sensitive about having their photo taken.)
- Shoot videos, too.
- Consider working in pairs or groups.
- Don't forget to ask basic questions about people's backgrounds – what kind of educational background do they have, what is their family structure like, how do they earn a living, how do they deal with a particular issue today, what do they spend on, etc.
- Use your eyes as much as your mouth and ears.

- Write down notes after each session – focus, in particular, on facts that you might forget later, as well as on unexpected things you discovered.
- As you begin to focus on a particular problem that you'd like to provide a solution for, ask more questions around that topic.

[Back to top.](#)

2) Prototype potential solutions.

- **Brainstorm.** Produce as many ideas as you can. Describe or sketch ideas on paper (or post-it notes), and lay them all out. During team brainstorming, make it a rule not to criticize each other's ideas. The goal is to generate as many ideas as you can, even if they're crazy.
- **Brainstorm more.** Considering combinations and variations of the solutions that you've come up with. Cluster ideas to see what comes up repeatedly. Sometimes, two crazy ideas combine make one really good idea. Still avoid criticizing.
- **Rank and select one or more ideas to prototype.** You'll want to make your decisions on some factors such as feasibility of the solution, degree to which it fits the constraints of the users, etc., and also on intuition that you might have. (Now, you can criticize!)
- **Prototype.** Depending on the complexity of the solutions, you may be able to prototype more than one solution. Having several prototypes can help determine what features work better than others.

[Back to top.](#)

3) User-test to see how the prototypes work or don't work.

- **Try your prototypes with potential users.** Work with your partner organization and/or with the community to do this. Depending on the nature of the problem and solution, this may require a lot of set up to implement. In some cases, you may not be able to test the solution fully. That's okay – in any case, try to find a way to present your ideas/prototypes to potential user groups.
- **Get feedback from users.** Again, learn as much as you can from users about their experience of the prototype. Again, talk to them, observe them, live with them. All of the guidelines in [Section \(1\)](#) still apply. In some cases, feedback may be indirect (someone may claim to like something, but then not use it at all) – pay attention!
- Compare your prototypes, if you have multiple ones.

- Try to understand as much as you can about why something is working or not working.

[Back to top.](#)

4) Iteratively prototype and test, repeating [steps 2](#) and [3](#).

- Repeat [steps 2](#) and [3](#) as much as you can.
- Combine the best features, if you have multiple prototypes.
- Go back to your earlier brainstorming and see if any old ideas might be useful.
- Brainstorm and prototype again, building on your newly gained understanding.
- Stop when you're happy that your solution is about as good as you can make it, or when you run out of time, whichever comes first.

[Back to top.](#)

5) Conduct a rigorous user study of your best solution. *(Optional, but recommended)*

This step is optional, but it can make a huge difference in the way that others (such as the Imagine Cup judges!) perceive the value of your project.

- **Design a study of your final solution that will allow you to compare it against some baseline.**
 - Figure out what kind of positive benefit you think your solution has. For example, students become better at long division with your solution.
 - Find a way to observe or measure it. Continuing the example above, you could administer a test that tests long division.
- **Set up a comparison.** There are several ways this can be done, but here are two common possibilities:
 - Create or identify two (or more) groups, communities, or environments where half of them use your solution, and the other half don't.
 - Set up a before-and-after study, where you observe.
- Try to maintain similarity between the compared cases in all respects, except for the fact of your solution being implemented.
- Run the study, and see if the outcomes match your expectations.

- **Incorporate the following tips...**

- Collect as much data as you can – use video recordings, if you can; interview participants before and after; record demographic information about the participants; run surveys or formal tests, etc.
- Again, get to know the user: Talk to them, observe them, live with them.
- Be as objective and fair as you can. Avoid biasing the results in any way.
- Many people consider a randomized control trial as the best way to test the value of a solution. See here for more details:
<http://www.povertyactionlab.org/research/rand.php>.
- During the study, do *not* change the solution. During the study, you may hit on a better way to do something, but resist the temptation to change the solution while the study is taking place, or you'll no longer be able to say exactly what it was you were testing.

[Back to top.](#)

Examples of User-Centered Technologies

The table below introduces just a few examples of user-centered technologies created by [Microsoft Research India's Technology for Emerging Markets group](#) that are the result of insights arrived at through close engagement with potential customers from underserved populations.

Research Insight	Solution	Description	Explore Further
<i>Many users are illiterate and/or have never used a computer or a mobile phone.</i>	Text-Free User Interfaces	Simple user friendly interfaces that use pictures, sounds and videos instead of written text.	Demo a Text-Free User Interface!
<i>Many farmers lack information on farming best-practices that can improve their farming outputs.</i>	Digital Green	A digital video database of farming best practices produced by local farmers and agricultural experts.	Watch rural farmers use Digital Green!
<i>In many classrooms PCs are shared by several students, and therefore do not allow for effective collaboration in learning activities.</i>	Windows MultiPoint	An application that enables multiple mice to be used on the same computer simultaneously allowing students to share computers effectively.	Watch a demo! Download the Windows MultiPoint SDK!
<i>SMS based services often require expensive GSM modems or the services of an SMS aggregator.</i>	SMS Server Toolkit	An SDK that allows developers to create and deliver SMS-based services with their PC!	Download the SMS Server Toolkit SDK!

Interested in viewing more examples of user-centered technologies?

- Visit mobileactive.org to learn more about how mobile phones are being leveraged for social impact!
- Search the [2008 Stockholm Challenge Project Database](#) for project descriptions and links to over 200 ICT projects with a strong user-centered design focus!

[Back to top.](#)

References

Here are some good papers that describe part of the process of UCD and user testing:

- MultiPoint
 - Initial exploration and prototyping: [Pawar, U. S., Pal, J., and Toyama, K. \(2006\) *Multiple mice for computers in education in developing countries*, IEEE/ACM Int'l Conf. on Information & Communication Technologies for Development, ICTD 2006](#)
 - A rigorous study: [Pawar, U.S., Pal, J., Gupta, R., and Toyama, K. \(2007\) *Multiple Mice for Retention Tasks in Disadvantaged Schools*, In Proceedings of ACM CHI'07, ACM Press](#)
- Text-Free UI
 - Initial exploration and prototyping: [Medhi, I., Sagar A., and Toyama K. *Text-Free User Interfaces for Illiterate and Semi-Literate Users*. International Conference on Information and Communication Technologies and Development. Berkeley, USA, May 2006](#)
 - A rigorous study of one aspect: [Medhi, I., Prasad, A. and Toyama K. *Optimal Audio-Visual Representations for Illiterate Users*. International World Wide Web Conference Committee. Canada, May 2007](#)

[Back to top.](#)