Notes

Chapter 2

1. LES contains four types of measures. Three ("benefits recognized early," "high benefits expected," and "direct elicitation of the construct") contain the core components of the lead user construct. The fourth ("applications generation") is a measure of a number of innovation-related activities in which users might engage: they "suggest new applications," they "pioneer those applications," and (because they have needs or problems earlier than their peers) they may be "used as a test site" (Morrison, Midgely, and Roberts 2004).

Chapter 3

1. Cluster analysis does not specify the "right" number of clusters—it simply segments a sample into smaller and smaller clusters until the analyst calls a halt. Determining an appropriate number of clusters within a sample can be done in different ways. Of course, it always possible to say that "I only want to deal with three market segments, so I will stop my analysis when my sample has been segmented into three clusters." More commonly, analysts will examine the increase of squared error sums of each step, and generally will view the optimal number of clusters as having been reached when the plot shows a sudden "elbow" (Myers 1996). Since this technique does not incorporate information on remaining within-cluster heterogeneity, it can lead to solutions with a large amount of within-cluster variance. The "cubic clustering criterion" (CCC) partially addresses this concern by measuring the within-cluster homogeneity relative to the between-cluster heterogeneity. It suggests choosing the number of clusters where this value peaks (Milligan and Cooper 1985). However, this method appears to be rarely used: Ketchen and Shook (1996) found it used in only 5 of 45 segmentation studies they examined.

2. http://groups-beta.google.com/group/comp.infosystems.www.servers.unix

3. http://modules.apache.org/

4. To measure heterogeneity, Franke and I analyzed the extent to which *j* standards, varying from [1; *i*], meet the needs of the *i* individuals in our sample. Conceptually, we first locate a product in multi-dimensional need space (dimensions = 45 in the case of our present study) that minimizes the distances to each individual's needs. (This step is analogous to the Ward's method in cluster analysis that also minimizes within cluster variation; see Punj and Stewart 1983.) The "error" is then measured as the sum of squared Euclidean distances. We then repeated these steps to determine the error for two optimally positioned products, three products, and so on up to a number equaling $I - 1$. The sum of squared errors for all cases is then a simple coefficient that measures how much the needs of *i* individuals can be satisfied with *j* standard products. The "coefficient of heterogeneity" just specified is sensitive both to the (average) *distance* between the needs and for the *configuration* of the needs: when the needs tend to form clusters the heterogeneity coefficient is lower than if they are evenly spread. To make the coefficient comparable across different populations, we calibrate it using a bootstrapping technique (Efron 1979) involving dividing the coefficient by the expected value (this value is generated by averaging the heterogeneity of many random distributions of heterogeneity of the same kind). The average random heterogeneity coefficient is then an appropriate value for calibration purposes: it assumes that there is no systematic relationship between the needs of the individuals or between the need dimensions.

5. Conceptually, it can be possible to generate "one perfect product" for everyone—in which case heterogeneity of demand is zero—by simply creating all the features wanted by anyone (45 + 92 features in the case of this study), and incorporating them in the "one perfect product." Users could then select the features they want from a menu contained in the one perfect product to tailor it to their own tastes. Doing this is at least conceptually possible in the case of software, but less so in the case of a physical product for two reasons: (1) delivering all possible physical options to everyone who buys the product would be expensive for physical goods (while costing nothing extra in the case of information products); (2) some options are mutually exclusive (an automobile cannot be both red and green at the same time).

6. The difference between actual willingness to pay and expressed willingness to pay is much lower for private goods (our case) than for public goods. In the case of private goods, Loomis et al. (1996) found the expressed willingness to pay for art prints to be twice the actual WTP. Willis and Powe (1998) found that among visitors to a castle the expressed WTP was 60 percent lower than the actual WTP. In the case of public goods, Brown et al. (1996), in a study of willingness to pay for removal of a road from a wilderness area, found the expressed WTP to be 4–6 times the actual WTP. Lindsey and Knaap (1999), in a study of WTP for a public urban greenway, found the expressed WTP to be 2-10 times the actual WPT. Neil et al. (1994) found the expressed WTP for conserving an original painting in the desert to be 9 times the actual WTP. Seip and Strand (1992) found that less than 10 percent of those who expressed interest in paying to join an environmental organization actually joined.

Chapter 6

1. As a specific example of a project with an emergent goal, consider the beginnings of the Linux open source software project. In 1991, Linus Torvalds, a student in Finland, wanted a Unix operating system that could be run on his PC, which was equipped with a 386 processor. Minix was the only software available at that time but it was commercial, closed source, and it traded at US$150. Torvalds found this too expensive, and started development of a Posix-compatible operating system, later known as Linux. Torvalds did not immediately publicize a very broad and ambitious goal, nor did he attempt to recruit contributors. He simply expressed his private motivation in a message he posted on July 3, 1991, to the USENET newsgroup comp.os.minix (Wayner 2000): *Hello netlanders, Due to a project I'm working on (in minix), I'm interested in the posix standard definition.* [Posix is a standard for UNIX designers. A software using POSIX is compatible with other UNIX-based software.] *Could somebody please point me to a (preferably) machine-readable format of the latest posix-rules? Ftp-sites would be nice.* In response, Torvalds got several return messages with Posix rules and people expressing a general interest in the project. By the early 1992, several skilled programmers contributed to Linux and the number of users increased by the day. Today, Linux is the largest open source development project extant in terms of number of developers.

Chapter 7

1. When they do not incorporate these qualities, they would be more properly referred to as networks—but communities is the term commonly used, and I follow that practice here.

2. **hacker** n. [originally, someone who makes furniture with an axe] 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. 2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming. 3. A person capable of appreciating **hack value**. 4. A person who is good at programming quickly. . . . 8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence *password hacker*, *network hacker*. The correct term for this sense is **cracker** (Raymond 1996).

3. Source code is a sequence of instructions to be executed by a computer to accomplish a program's purpose. Programmers write computer software in the form of source code, and also document that source code with brief written explanations of the purpose and design of each section of their program. To convert a program into a form that can actually operate a computer, source code is translated into machine code using a software tool called a compiler. The compiling process removes program documentation and creates a binary version of the program—a sequence of computer

instructions consisting only of strings of ones and zeros. Binary code is very difficult for programmers to read and interpret. Therefore, programmers or firms that wish to prevent others from understanding and modifying their code will release only binary versions of the software. In contrast, programmers or firms that wish to enable others to understand and update and modify their software will provide them with its source code. (Moerke 2000, Simon 1996).

4. See www.gnu.org/licenses/licenses.html#GPL

5. http://www.sourceforge.net

6. "The owner(s) [or 'maintainers'] of an open source software project are those who have the exclusive right, recognized by the community at large, to *redistribute modified versions*. . . . According to standard open source licenses, all parties are equal in the evolutionary game. But in practice there is a very well-recognized distinction between 'official' patches [changes to the software], approved and integrated into the evolving software by the publicly recognized maintainers, and 'rogue' patches by third parties. Rogue patches are unusual and generally not trusted." (Raymond 1999, p. 89)

Chapter 8

1. See also Bresnahan and Greenstein 1996b; Bresnahan and Saloner 1997; Saloner and Steinmueller 1996.

Chapter 10

1. ABS braking is intended to keep a vehicle's wheels turning during braking. ABS works by automatically and rapidly "pumping" the brakes. The result is that the wheels continue to revolve rather than "locking up," and the operator continues to have control over steering.

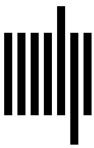
2. In the general literature, Armstrong's (2001) review on forecast bias for new product introduction indicates that sales forecasts are generally optimistic, but that that upward bias decreases as the magnitude of the sales forecast increases. Coller and Yohn (1998) review the literature on bias in accuracy of management earnings forecasts and find that little systematic bias occurs. Tull's (1967) model calculates $15 million in revenue as a level above which forecasts actually become pessimistic on average. We think it reasonable to apply the same deflator to LU vs. non-LU project sales projections. Even if LU project personnel were for some reason more likely to be optimistic with respect to such projections than non-LU project personnel, that would not significantly affect our findings. Over 60 percent of the total dollar value of sales forecasts made for LU projects were actually made by personnel not associated with those projects (outside consulting firms or business analysts from other divisions).

Democratizing Innovation

By: Eric von Hippel

Citation:

Democratizing Innovation
By: Eric von Hippel
DOI: 10.7551/mitpress/2333.001.0001
ISBN (electronic): 9780262285636
Publisher: The MIT Press
Published: 2006



The MIT Press

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please email special_sales@mitpress.mit.edu or write to Special Sales Department, The MIT Press, 5 Cambridge Center, Cambridge, MA 02142.

Set in Stone sans and Stone serif by The MIT Press. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Hippel, Eric von.
Democratizing innovation / Eric von Hippel.
p. cm.
Includes bibliographical references and index.
ISBN 0-262-00274-4
1. Technological innovations—Economic aspects. 2. Diffusion of innovations.
3. Democracy. I. Title.
HC79.T4H558 2005
338'.064—dc22 2004061060

10 9 8 7 6 5 4 3 2 1