DEVELOPER'S DISEASE CAN BE HAZARDOUS TO YOUR HEALTH

A developer's inability to acknowledge bad news about a project can lead to business disaster.

by David C. Bamberger

Real estate developers typically are an optimistic lot. Get a group of developers together, and you will hear little bad news. Even in today's turbulent economic climate, developers are upbeat. In Colorado, developers already are talking about a market turnaround. Real estate developers are like skilled rock climbers who can spot a viable route to the top of even the most obscure and difficult crag. They see opportunities long before the future has come into focus for most of the rest of us. The developer's imperative is, "I build; therefore, I am."

Many of us believe that optimism is one of the necessary ingredients for success in the development business. It is important to be positive about the market when talking to anyone about a proposed development project. Can you imagine a developer telling his lender nothing but bad news about a proposed project and its prospects for success? Developers must be effective salesmen, and the best sales strategy is to emphasize the good points and hide the bad points, right?

Wrong. As important as optimism is to developers, it can be one of the most likely causes of their failure. Combine optimism with someone else's money, and you have the ingredients for a potential business disaster. The recent failure of many of the savings and loan institution-financed real estate projects is a good example. Optimistic developers, skilled at selling development projects, teamed up with the S&L money machine in a classic case of "skilled incompetence." Despite the best of intentions of many of the players, the projects had disastrous, unplanned outcomes.

Developer's Disease

In my years of experience as a consultant, I frequently have encountered a phrase that characterizes the "skilled incompetence" of some developers. The phrase is "developer's disease," and it reflects a developer's inability to look objectively at market and financial facts and act on those facts accordingly. A developer with developer's disease typically brushes aside any information that suggests his proposed project may not work and then unintentionally covers up the fact that the information even exists.

What is really behind this malady called "developer's disease?" Several researchers, including Chris Argyris, a Harvard University professor, have demonstrated that human beings are not very good learners, especially when they are dealing with threatening situations.[1] Research has shown that, when faced with difficult issues, people protect themselves from threat by applying action strategies that involve:

- looking for evidence to support their views
- ignoring evidence that indicates they may be wrong

David C. Bamberger *is an independent consultant with a practice in applied economics and real estate research. He is affiliated with Joseph Farber and Co., Inc., a real estate appraisal and consulting firm in Denver, Colorado.*

- skewing inferences in self-protective directions
- failing to test their views publicly
- discouraging any open discussion

A developer uses these same strategies to protect himself from the threat of losing his project. Putting together a development project is no easy task. Acquiring land, drawing up plans, securing financing and obtaining the necessary land use entitlements and permits takes many months, sometimes even years, and requires strong advocacy of the project's merits by the developer. Good news sells development projects in today's approval system; bad news does not. It is no wonder that developers must be optimistic to survive.

This same optimism that leads to the success of one project can lead to the failure of another. The strategies used by the developer to protect a project from being killed before it gets off the ground are "anti-learning" because they close the developer's mind to facts and prevent him from learning. These behaviors generally lead to what is called "self-sealing logic" which is employed by people who think things are true simply because they wish them to be true.

A developer who has developer's disease is not difficult to spot; he is always selling his project with very persuasive arguments and elaborate dialog. In order to protect his project from any bad news, he does not test his reasoning about the project's true merits either privately or publicly. Some developers are so heavily affected by developer's disease that their staff are afraid to bring them any bad news. Optimism about the project is the norm, which is scrupulously enforced around the office. Consequently, there is no talk of the negatives. Staff often thinks the developer knows something about the merits of the project that they do not. Staff also is unwilling and unable to test the developer's knowledge. They avoid upsetting the developer by not asking difficult questions and discussing bad news.

Productive Reasoning

Fortunately, not every developer has developer's disease; in fact, many seem to be immune from it. I have known and worked with a number of developers who do not enter the marketplace with poorly timed and ill-planned projects. Many of these people are in business today despite the difficult time the industry has faced over the past cycle. What do these developers have that others do not?

Successful developers employ "productive reasoning" to provide sure-fire protection from the disease. Productive reasoning is not a new concept. Most of us claim to use it in business and even in our daily lives. The ingredients of productive reasoning are:

- collecting and using hard data
- reasoning open to public inspection
- connecting conclusions with data
- publicly testing inferences and conclusions

The process of productive reasoning involves a chain of activities that starts with the collection of hard data. To make sense of the data, models are built, and an analysis is performed. This analysis may involve simple, back-of-the-envelope techniques or powerful and sophisticated quantitative methods. Inferences about the data and analysis are then drawn, and conclusions are developed that lead to action.

Productive reasoning is not a purely data-driven process; there is more to it than just science. Successful developers implement productive reasoning through the following action strategies:

- searching for and acting on disconfirming information
- continuously checking logic both publicly and privately
- remaining open to constructive confrontation
- encouraging others to test their reasoning
- considering mistakes as part of learning

Productive reasoning is built on a foundation of hard data, clear thinking and continuous testing of inferences and conclusions. Developers who employ productive reasoning govern their actions by acquiring and acting on valid information. They do not become mired in a position of advocacy but are always on the lookout for data that may prove them wrong. Developers who employ productive reasoning encourage their staff to confront their logic, to dig for the facts, to constantly question the project's merits and continuously search for the right strategy to make it work. Developers who employ productive reasoning are not afraid to modify their position and their project in the service of learning; they create an atmosphere of inquiry around the office.

Defensive Reasoning

The opposite of productive reasoning is defensive reasoning, the primary cause of developer's disease. Defensive reasoning uses soft data or no data at all, and the reasoning process is kept hidden in a black box, unavailable for public inspection and testing. Developers who employ defensive reasoning discourage any serious questioning in order to protect themselves from the possibility that they may be wrong; they do not discuss bad news around the office; they seal themselves from the reality of the marketplace and conclude that, "If we build it, the market will come."

Conclusion

A chronic case of developer's disease is usually terminal. Although productive reasoning is not a guarantee that a developer will fluorish in the turbulent 1990s, it will increase his chances for survival. The lesson for anyone with even a mild case of developer's disease is to learn to adopt productive reasoning before it is too late.

NOTE

1. Argyris, Chris, Putnam, Robert and Smith, Diana. *Action Science* (San Francisco: Jossey Bass, 1985).