3

Responses to Change

According to the Satir Change Model, change happens one person at a time, and each person or organization has many choice points—many points at which several responses are possible. The cumulative effect of these choice points creates the change, which—in spite of appearances—doesn't come suddenly. These are the points of great interest, because they are the points at which the change process can be managed. This chapter will examine some of the factors that influence the choice of response.

3.1 Choice Points

Figure 3-1 is an overall diagram of the Satir Change Model, emphasizing choice points (in rounded rectangles):

- The foreign element can be rejected, or not rejected.

- The foreign element can be accommodated into the old model of reality.
- The old model can be transformed to receive the foreign element.
- The transformation can be integrated or not integrated into the model.
- The transformed model can be mastered or not mastered through practice.
- In addition, there is the choice of how much time should pass before the explicit introduction of a new foreign element, though some foreign elements don't give us that choice.

Let's work through examples of each, using the attempted introduction of object-oriented technology as the foreign element in question.

3.1.1 Rejecting the foreign element

When management announces that a project will use an object-oriented methodology, the managers want to ensure that this approach is accepted. Developers, however, may perceive it as a foreign element, and many of them may attempt to reject it by such actions as

- not seeing the announcement memo in their e-mail.
- proving that an object-oriented approach cannot be used on their module.
- proving that an object-oriented system would be hopelessly inefficient.
- proving that changing approaches would take too long.
- forgetting to attend required classes.
- making halfhearted attempts, then becoming discouraged when they don't work, saying "It was a stupid idea anyway."
- producing a working version the old way, then rationalizing, "We might as well use it."

Some of these actions are simply passive-aggressive, while others contain potentially valid arguments. Although arguments don't have to be valid to be used in an attempt to reject a foreign element, the possibility that they are valid makes them harder to counter. Fortunately, the job of management is not to counter arguments, but to get the job done, so the first step toward success is to stop arguing and realize that opposition to a foreign element is perfectly natural, and not a personal attack.

Once managers understand it's not personal, they'll be in a position to listen to the sense of each argument and, more importantly, to the emotional "music" behind it. By responding to the emotions, managers will generally be more successful than trying to counter the arguments. Of course, if "object-

oriented
they ha
sonal.

Figure 3-1.

oriented" is just a disguised symbolic battleground for "who's in charge," then they have a different situation entirely—and the attacks may very well be personal.
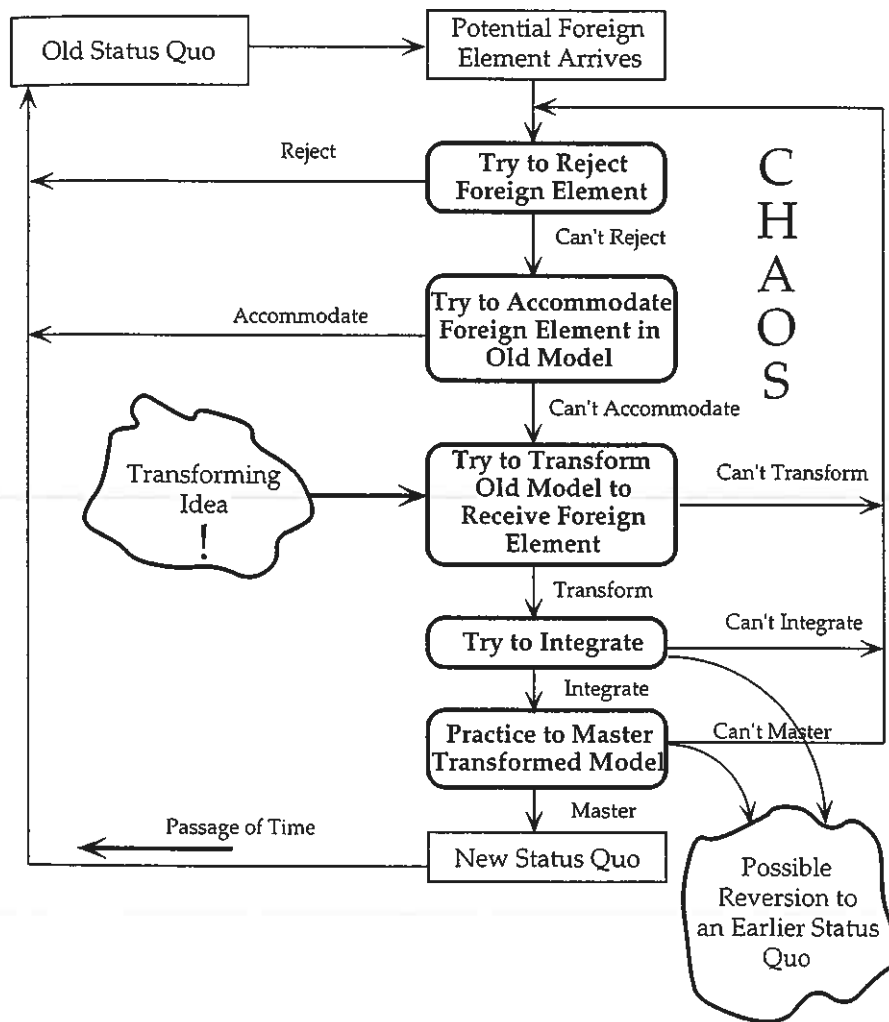

Figure 3-1.   According to the Satir Change Model, there are many choice points that can undermine or support the change process.

3.1.2 Accommodating the foreign element into the old model

If these attempts to reject the new methodology don't succeed, a developer may resort to accommodating the foreign element into the old model by such actions as

- claiming that the COBOL code is "actually object-oriented, more or less"
- doing everything the same old way, but adding a step at the end to translate code into something that will be compiled by the object-oriented compiler

Because the foreign element is accommodated into the old model, the developers truly believe they are doing object-oriented development, so it's never a good idea to accuse them of passive-aggression. Instead, a good strategy is to have an experienced and tactful person work with them "to improve their use of object-oriented development."

3.1.3 Transforming the old model to receive the foreign element

The tactics of rejection or accommodation to the old model may actually be tried by the developers, or they may simply be tried in their heads. In either case, if they fail to get rid of the foreign element and go back to the Old Status Quo, the developers may then try to transform their old models.

At this stage, the purveyors of change make one of their most common and devastating mistakes. Instead of helping the developers see ways in which an object-oriented approach resembles what they already know, they emphasize how everything is entirely new and different. One developer told me of her experience in an introductory object-oriented programming course:

> On the first morning of the first day, the instructor, who wore a suit and tie (which no developer in our company ever wears) and looked about seventeen years old, stood up in front of the class and declared, "The first thing you must do is forget everything you ever thought you knew about programming. I will not answer questions about how object-oriented programming is like anything else, because it is not like anything else, so please don't waste my time." He then told us his name.
>
> I completely tuned out, and the only other thing I clearly remember was when he said he was the principal consultant to our company on the change to object-oriented programming, so he would be working with us in the coming weeks to make it a reality. I remember laughing to myself and thinking, "Fat chance!" I don't know if anyone else had the same reaction I did, because nobody ever talked about the course, the instructor, or object-oriented programming. I do recall seeing him in the corridors once or twice, but then he was gone, and I hadn't thought about him again until just now.

From a distance, it's easy to recognize that this brash young instructor probably didn't *know* anything about any other approach to programming, and so had to protect himself by prohibiting questions. The new messiah should have had the knowledge and inclination to show people that they really had a vast amount of knowledge about programming, and that object orientation was only a small, logical increment to that knowledge base. For example, a good way to do this is for the instructor to encourage the developers to rethink some of the tough problems they have previously solved, but use the new paradigm as an added tool.

3.1.4 Integrating the transformation

The introduction of object-oriented methods often fails at the point where the new way must be integrated into practice. Some people just don't get it, but many get it in class and then lose it when they try to create examples. That's why workshop classes are more effective when they use real examples.

Few workshop instructors take the trouble to generate *real* examples, which is not the same as spending hours formatting and polishing an artificial example that would be easy to understand. Quite often, the problem here is a kind of Newtonian rush—pressure to get it not just correct, but fast. This is the typical classroom mentality, where students are ridiculed because they didn't complete an assignment in the allocated thirty minutes, especially if it's a schoolbook example that the instructor has worked out in advance. The effective instructor establishes a no-ridicule environment, and is willing to work privately with each student until they've all had a taste of personal success to start them on the way to Integration.

Even when the instructor does a good job, the same pressure for speed is seen when the novice is expected to return from a class and immediately use the new approach on an ongoing project—with an increase in productivity, and certainly no loss, an expectation that flies in the face of the temporary loss predicted by the Satir Change Model. A better management approach is to have the novice sit alongside an experienced person, watching a real example unfold, then try parts of an example while being observed by the experienced person. If no experienced person is available, you'll have to plan a lot of extra time and variability for learning, so find an experienced person.

In either case, the pressure to go quickly in an unsafe environment often leads sincerely dedicated people to fail to integrate the new idea, and they return to Chaos, thus aborting the change effort. A change of this magnitude requires a vast approach, with much more than dipping the sheep in a class or two. If instead, we get a half-vast approach, people fail to integrate the new way. Worse than that, the next half-vast approach to introduce object-oriented methods is met by the rejection, "We tried that already, and it didn't work."

3.1.5 Mastering the transformed model

Once the change has been integrated into a few working examples and the system enters the New Status Quo stage, a return to Chaos becomes far less likely—but still possible if conditions are bad enough. Though the transforming idea may be monumental, lots of minor adjustments are required to make it work in practice. If managers lack patience with details or, worse, denigrate those who implement those details, the monument may crack on the first try, like the Liberty Bell.

A second common shortcoming in the New Status Quo stage is the failure to allow for scaling-up from small examples. Object-oriented techniques are an archetypical example of this dereliction. With a change this large, I like to move my clients gradually from half-hour classroom examples to two-hour applications, followed by one-day applications, then one-week applications, before doing a significant pilot project of a month or longer. This expenditure of time may render the Hole-in-the-Floor gang despondent, but if the change doesn't justify the expense, perhaps you shouldn't be doing it in the first place.

3.1.6 Timing

Perhaps the most common cause of failing to change is the question of timing—as seen in the interference from other changes. Changes do not come in isolation. Often, we are hit by a second change long before we have reached New Status Quo on the previous one. Or, changes may be few and far between, with long gray periods resting in Old Status Quo. In any case, managers who wish to introduce change will want to know how often they can reasonably introduce foreign elements. For guidance, they can use the Zone Theory.

3.2 Timing Change Interventions with McLyman's Zone Theory

To deal with the timing of change, Lynda McLyman of Progress Associates marked out four zones in the Satir Change Model, as shown in Figure 3-2. According to this model, we receive change opportunities differently, depending upon the zone in which we find ourselves.[3]

3.2.1 The Red Zone

The Red Zone is the time before a previous foreign element is transformed, accommodated, or rejected. When a new foreign element arrives while the system is in the Red Zone, chaos from both foreign elements increases. Moreover, the chance of ever finding a transformation for either foreign element decreases, and the likelihood of rejection or accommodation increases.

In short, the change that might have been stimulated by the foreign element is much less likely to take hold. If a number of Red Zone cycles occur in

(marginal text, partially cut off)

quick

produ

curren

multip

people

stallio

dizzy

Performance

Figure 3

3.2.2 1

The Ye

being i

the Yel

as witl

nal tra

M

eign el

quick succession, the system may get stuck in Chaos and become totally non-productive. Multiple foreign elements in the Red Zone *could* also lead to concurrent changes and synergy, where the transforming idea accommodated multiple concurrent changes. This is the kind of "holy grail" idea that makes people sit around immobilized waiting for Sir Lancelot to arrive on a white stallion. They shouldn't hold their breath! After a while, they'll start getting dizzy and seeing illusions—making Hatchet Man look like Sir Lancelot.
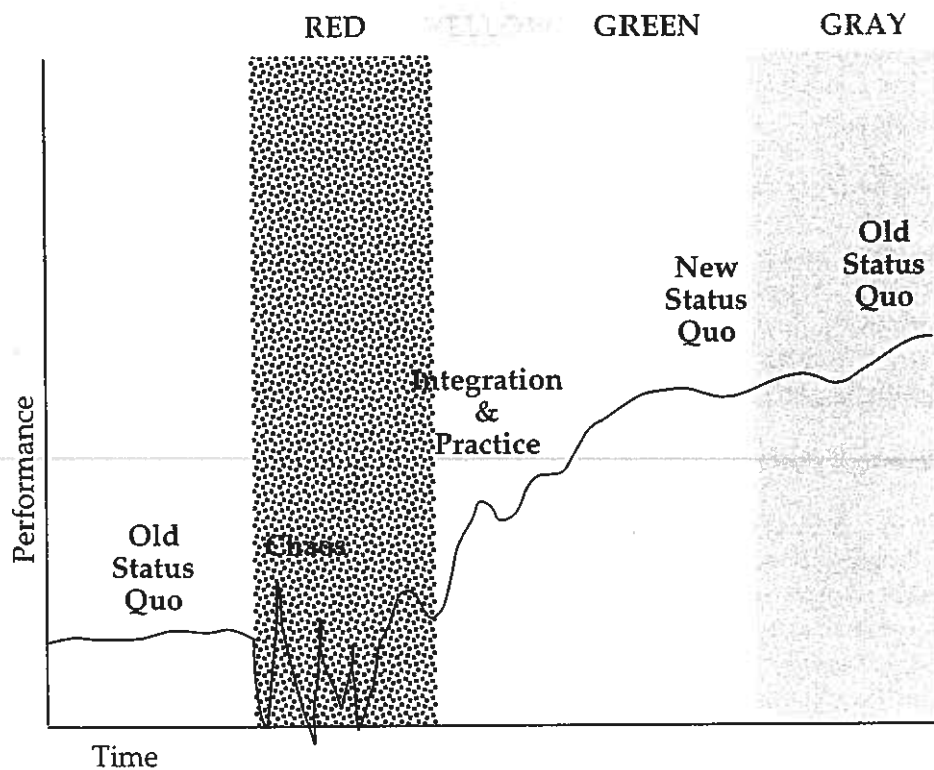

Figure 3-2.   McLyman's Zone Theory predicts that responses to a new foreign element will differ according to where the system or individual is in the current change cycle.

3.2.2 The Yellow Zone

The Yellow Zone is the time interval when a previous transformation is still being integrated. When a new foreign element arrives while the system is in the Yellow Zone, chances of successful change are reduced, but not as seriously as with Red Zone foreign elements. The system may lose its grip on the original transforming idea and be thrown back into Chaos.

More important is the effect over time. With successive Yellow Zone foreign elements, the system builds an *energy debt:* Successful change becomes

less and less likely, and productivity drags. After four or five Yellow Zone foreign elements, the system loses all chance of successful change and slips into Chaos. After that, further attempts to change will create Red Zone effects, and breakdown is close at hand.

3.2.3 The Green Zone

The Green Zone is the time between late Integration and early New Status Quo. When a foreign element arrives in the Green Zone, the system's chances of successful change are maximized. Not only is there no energy debt, but each successful Green Zone change increases the chances for the next. There has been sufficient time to enjoy the successful change and to recharge the emotional batteries.

3.2.4 The Gray Zone

The Gray Zone is all the time after the system has been in Late Status Quo for a while. When a foreign element arrives in the Gray Zone, people have already lost some of their meta-change skills, for old learnings about change have lost their usefulness. Without these meta-change skills, change is once again slow and difficult, and the chance of successful change declines.

3.2.5 Lessons for managers

McLyman's Zone Theory contains several lessons for those who would manage change. First, there are the obvious lessons of timing. Managers who are in a hurry and press the organization with too many changes too quickly will merely slow down the very changes they are trying to accelerate. Similarly, if managers adopt the strategy of "hit them with a lot of changes, and some will stick," they'll find that in the end, none of them will stick. Or, worse, the wrong ones will stick.

The Yellow Zone holds particular caution for managers who attempt these strategies. A few hits in the Yellow Zone may happen to succeed, thus encouraging a manager to keep piling on foreign elements, much like a gambler who happens to win the first few bets.

Most managers are aware of the Gray Zone and try to stimulate the organization with frequent changes. What they may fail to notice, however, is that parts of the organization are not touched by these changes, and are deep in the Gray Zone in an otherwise Green Zone organization.

3.2.6 One person at a time

Perhaps the most important lesson of the Zone Theory is that not all parts of the system are in the same zone at the same time. This is true at every level of the organization, right down to the individual. Foreign elements don't segre-

gate according to parts of our lives. Regardless of their origin, they all add together. Thus, in a department that is generally in the Green Zone, there may be one individual who has been experiencing a large number of personal foreign elements, and may be deeply in the Red Zone.

In the same Green Zone department, a few individuals may be largely untouched by the changes all around. These people may be in their personal Gray Zone, and simply not know how to cope with change when their turn arrives.

Most books and courses on change management emphasize the need for strategic planning. The Zone Theory reminds us that although change must be managed at a high level, we must never ignore the impact on individuals. As Satir said, "Change happens one person at a time."

3.3 Patterns of Information Flow

Obviously, managers need a continuing flow of information to make successful interventions in change. Unfortunately, change tends to disrupt information flow. The Satir Change Model helps us to understand what each stage does to information feedback mechanisms. It also shows us that during change, the most reliable information is the *emotional signals from the people experiencing the change*. You can use these signals, for instance, to determine the appropriate zone strategy, or what kind of information you need to supply.

3.3.1 Old Status Quo

During an aging Status Quo stage, old feedback mechanisms are eroding slowly. Information is not getting through. As the system starts breaking down, behavior becomes less predictable, and to make it more predictable, people often ignore what information does get through. For example, as the number of trouble incidents from the field grows, managers may compare them to the previous week. That way, the increase may be lost in the noise or labeled not so bad.

In the Old Status Quo stage, interventions should be designed to get people to recognize what is, rather than what they've grown accustomed to seeing. To evaluate the current number of trouble incidents, managers should compare them to the average performance in the past, so as to reveal—rather than conceal—any long-term upward trend.

3.3.2 Foreign element

When the foreign element arrives, the old feedback mechanisms may fail completely. Very often, the arrival of the foreign element is merely a breaking through of the illusions created by the old feedback mechanisms. The best interventions here are to help people stay with the information and believe it,

even though believing it will send them into Chaos. For example, the Public Project Progress Poster (PPPP) places information about schedule slippage right out in front, where everyone can see what it is, and if it's being manipulated.[4]

It's especially important not to let the system punish those who bring the new information. The PPPP gives this type of protection by making schedule updates a routine matter, not subject to choice and thus not subject to intimidation.

3.3.3 Chaos

In Chaos, the old feedback mechanisms are shattered. The system runs wild, and people are unable to reconnect with any model of what's happening to them. People desire stability so greatly that they may attach themselves to any source that appears to know what's happening, from tea leaf readers to con artists to methodology vendors. What's needed here is persistent, compassionate offers of reliable information about what's really happening, not the often brutal, so-called facts that will destroy, but also not the kind of placating that will allow temporary return to the status quo: "We're losing nine customers per month . . . but that's not *too* bad. We could be losing ten per month."

Continued cycles of group discussions and individual experimentation will help ensure that people are listening. The only schedules that should be maintained are schedules driven by the need for learning, such as getting people to sources of new information on time.

3.3.4 Integration and Practice

During Integration, new arrangements of feedback mechanisms begin to appear. As they are tested in Practice, some order evolves. At times, because the new mechanisms are not well developed, feedback is slower than needed and this causes the system to oscillate wildly. At this point, people can be helped with specific techniques for getting information, such as how to read their emotional state more quickly and reliably. Managers must try to create a climate in which it's okay even for senior technical leaders not to know things, and to ask questions about those things. Because the goal is merely to get control of the learning process, schedules should contain lots of slack. The payoff in speedy performance will come later, when the New Status Quo is reached.

3.3.5 New Status Quo

As the system moves into the New Status Quo, new mechanisms are in place and working rather well. Practice brings improvement, but appropriate information flows freely to and from the system. In this stage, what people need most is the permission to make mistakes—to be honest so that they can explore