

## Chapter 20

# TEAMICIDE

What's called for here is a concise chapter entitled "Making Teams Jell at Your Company." It should have half a dozen simple prescriptions for good team formation. These prescriptions should be enough to guarantee jelled teams. In the planning stage of this work, that is exactly the chapter we expected to write. We were confident. How difficult could it be to cut to the heart of the matter and give the reader practical tools to aid the process of making teams jell? We would apply all our skills, all our experience; we would overwhelm the problem with logic and pure brilliance. That's how it looked in the planning stage. . . .

Between plan and execution, there were a few distressing encounters with reality. The first of these was that we just couldn't come up with the six prescriptions needed for the chapter. We got stuck at zero. We'd been prepared to scale our expectations down a bit, but not this much. ("Zero Things You Can Do to Make Teams Jell"?) It seemed clear that something was wrong with the underlying notion of the chapter. What was wrong was the whole idea of making teams jell. You can't make teams jell. You can hope they will jell; you can cross your fingers; you can act to improve the odds of jelling; but you can't make it happen. The process is much too fragile to be controlled.

Part of our scaling down of expectations involved a change in vocabulary. We stopped talking about *building* teams, and talked instead of *growing* them. The agricultural image seemed right. Agriculture isn't entirely controllable. You enrich the soil, you plant seeds, you water according to the latest theory, and you hold your

breath. You just might get a crop; you might not. If it all comes up roses, you'll feel fine, but next year you'll be sweating it out again. That's pretty close to how team formation works.

Back to brainstorming mode: We began looking for "Six Things You Can Do to Make Team Formation Possible." It was still hard. At last, in desperation, we tried a trick called *inversion*, described in Edward deBono's *Lateral Thinking*. When you're stuck trying to solve a problem, deBono suggests that rather than looking for ways to achieve your goal, look for ways to achieve the exact opposite of your goal. This can have the effect of clearing away the brain's cobwebs that keep you from being creative. So instead of looking for ways to make team formation possible, we began to think of ways to make it impossible. That was easy. In no time at all, we came up with lots of sure-fire ways to inhibit the formation of teams and disrupt project sociology. These measures, taken together, constitute a strategy we dubbed *teamicide*. Our short list of teamicide techniques is presented below:

- defensive management
- bureaucracy
- physical separation
- fragmentation of people's time
- quality reduction of the product
- phony deadlines
- clique control

Some of these techniques will look awfully familiar. They are things that companies do all the time.

## Defensive Management

It makes good sense for you the manager to take a defensive posture in most areas of risk. If you must work with a piece of failure-prone gear, you get a backup; if the client is inclined to vacillate, you take pains to nail down the product specifications; if a contract vendor tends to "forget" promises, you publish minutes after each meeting.

There's one area, though, where defensiveness will always backfire: You can't protect yourself against your own people's incompetence. If your staff isn't up to the job at hand, you will fail. Of course, if the people are badly suited to the job, you should get new people. But once you've decided to go with a given group,



your best tactic is to trust them. Any defensive measure taken to guarantee success in spite of them will only make things worse. It may give you some relief from worry in the short term, but it won't help in the long run, and it will poison any chance for the team to jell.

*I found myself one day giving Consultant's Speech Number 9B to a project group, chastising them because they'd failed to get client approval for their emerging concept of a new system. They all looked faintly embarrassed. Finally one of them said, "We all agree that the client ought to be seeing this stuff. But our boss has laid down a firm rule that nothing will ever be shown to people outside the project unless it has his approval." She went on to explain that the boss was so swamped that months of work had piled up in his inbox. What option did they have? They were just plugging away in the dark knowing full well that most of what they were producing wouldn't pass muster with the client staff when it was finally shown to them.*

—TRL

The boss didn't trust his own people. He was worried they might show something that was wrong to client personnel. He was worried that their errors might reflect badly on him. Only his own judgment was competent; anyone else's was suspect.

If you're the manager, of course you're going to feel that your judgment is better than that of people under you. You have more experience and perhaps a higher standard of excellence than they have; that's how you got to be the manager. At any point in the project where you don't interpose your own judgment, your people are more likely to make a mistake. So what? Let them make some mistakes. That doesn't mean you can't override a decision (very occasionally) or give specific direction to the project. But if the staff comes to believe it's not allowed to make any errors of its own, the message that you don't trust them comes through loud and clear. There is no message you can send that will better inhibit team formation.

Most managers give themselves excellent grades on knowing when to trust their people and when not to. But in our experience, too many managers err on the side of mistrust. They follow the



basic premise that their people may operate completely autonomously, as long as they operate correctly. This amounts to no autonomy at all. The only freedom that has any meaning is the freedom to proceed differently from the way your manager would have proceeded. This is true in a broader sense, too: The right to be right (in your manager's eyes or in your government's eyes) is irrelevant; it's only the right to be wrong that makes you free.

The most obvious defensive management ploys are prescriptive Methodologies ("My people are too dumb to build systems without them") and technical interference by the manager. Both are doomed to fail in the long run. In addition, they make for efficient teamicide. People who feel untrusted have little inclination to bond together into a cooperative team.

## Bureaucracy

Studies conducted by Capers Jones in the 1970s and 1980s reported on systems development costs by work category. One of the categories was "Paperwork." What Jones calls *paperwork* is more or less mindless paper pushing, since the thinking time necessary to decide what to put on the paper is categorized as some other activity, such as analysis, design, or test planning. In other words, his "paperwork" category is pure bureaucracy. Jones concluded that paperwork is the second largest category of systems development work. It accounts for more than thirty percent of the cost of producing a given product.

There is a depressing modern trend to make development workers more and more into bureaucrats. Perhaps this is a sign of epidemic defensive management. But while the trend is global, it is not at all uniform. We know of companies in which the development groups look and feel like a bureaucratic nightmare by Kafka, and other companies in which the paperwork burden is minuscule.

Mindless paper pushing is a waste. It ought to be attacked because it keeps people from working. But our point here is a slightly different one. It is that bureaucracy hurts team formation. The team needs to believe in whatever goal it forms around. That goal can be arbitrary, but at least it has to exist. There has to be some evidence that management believes in it. Just telling your people that the goal matters won't be enough if you also have to tell them they should spend a third of their time pushing paper. Paper pushers just can't get themselves into SWAT Team mode. They can't see themselves hellbent for success.



## Physical Separation

When the Furniture Police makes its case for the Zippo-Flippo Modular Office System, all the talk is of "flexibility." But when it comes time to flex a bit to put a work group together, the long faces come out. "We can't disrupt everything and move stuff around over our lovely carpet just to get these four people into adjacent space. Can't they use telephones?" The result is that what could be a tightly bound team is scattered over multiple floors or even in different buildings. The specific work interactions may not suffer terribly, but there is no casual interaction. Group members may grow stronger bonds to nongroup neighbors, just because they see more of them. There is no group space, no immediate and constant reinforcement, no chance of a group culture forming. (You couldn't imagine Black Team members all dressing in black if their workspaces were not together; they'd be constantly interacting with people who weren't in on the joke, who just thought they were bizarre, and the whole funny bit would die with a whimper.)

Physical separation of people who are expected to interact closely doesn't make much sense anyway. Neighboring workers are a source of noise and disruption. When they're all on the same team, they tend to go into quiet mode at the same time, so there is less interruption of flow. Putting them together also gives them opportunity for the casual interaction that is so necessary for team formation.

## Fragmentation of Time

*One of my clients is an agency of the Australian government. During one consulting call, I collected data indicating that the average worker there was involved in four or more different projects. I complained about this to the Commissioner. He said it was regrettable, but just a fact of life. People's duties were fragmented because their skills and knowledge made them indispensable to efforts other than the principal ones they were assigned to. He said it was inevitable. I said it was nonsense. I proposed that he make it a specific policy that people be assigned to one and only one project at a time and that the policy be written down and widely distributed. He was game. A*



*year later, when I returned, the average worker was assigned to fewer than two projects.*

—TDM

Fragmentation is bad for team formation, but it's also bad for efficiency. (Perhaps you've begun to pick up a trend here.) People can keep track of only so many human interactions. When they try to be part of four working groups, they have four times as many interactions to track. They spend all their time changing gears.

No one can be part of multiple jelled teams. The tight interactions of the jelled team are exclusive. Enough fragmentation and teams just won't jell. The saddest thing is we allow far more fragmentation than is really necessary. We tend to concede this battle without even a fight. Simply saying that a goal is to assign people only one piece of work at a time can result in significant reductions of fragmentation, and thus give teams a real chance to form.

### **The Quality-Reduced Product**

The heading used here is a facetious one; nobody really talks about quality-reduced products. What they talk about is cost-reduced products. But it usually boils down to the same thing. The typical steps we take to deliver a product in less time result in lower quality. Often the product's end user gives willing consent to this trade-off (less quality for earlier, cheaper delivery). But such concessions can be very painful for the developers. Their self-esteem and enjoyment are undermined by the necessity of building a product of clearly lower quality than what they are capable of.

An early casualty of quality reduction is whatever sense of team identification the group has been able to build. Co-workers who are developing a shoddy product don't even want to look each other in the eye. There is no joint sense of accomplishment in store for them. They know that there will be a general sense of relief when they can stop doing what they're doing. At the end of the project, they'll make every effort to separate themselves from other members of the group, and get on to better things.

### **Phony Deadlines**

In Chapter 3, we made the point that tight deadlines can sometimes be demotivating. But there are certainly cases where a tight but not



impossible deadline can constitute an enjoyable challenge to the team. What's never going to help, however, is a *phony* deadline. When the manager intones, "We absolutely must be done by \_\_\_\_\_," group members can barely keep their eyes from rolling. They've been there before. They know the whole routine.

Maybe phony deadlines used to work. Maybe there were once workers so naive that they actually believed what they heard. When the boss said the job "reeely reeely has to be done by January," maybe they just accepted it and buckled down. Maybe. But it certainly doesn't work that way anymore. The people on your staff will know if they're being bamboozled. If you say the product absolutely has to be out the door by some arbitrary date, they will ask, "Why? Will the universe grind to a halt if we're late? Will the company fold? Will the nation slide into the sea? Will Western Civilization break down?"

In the typical phony deadline spiel, the manager announces that the work must be done on such and such a date. The date mentioned is impossible to meet, and everyone knows it. The effort will certainly slip (so much for the idea that the deadline is absolute). The work has been defined in such a way that success is impossible. The message to the workers is clear: The boss is a Parkinsonian robot with no respect or concern for them. The boss believes they won't do a stroke of work except under duress. Don't expect a jelled team on that project.

## Clique Control

A participant at one of our seminars made this observation: "The only time our management shows any awareness of teams is when it takes specific steps to break them up." There may be an explicit policy that teams can't be allowed to stay together from one job to another. Or, there may be a policy that projects winding down have to be de-staffed smoothly over time so that the personnel organization can steer people efficiently into new projects. This assures that teams will be broken up. Still other organizations take no specific steps to disband teams, but miss every opportunity to keep them together.

The pleasures of team activity and the energy that is produced by team interaction are articles of faith in our society. How did business organizations ever come to be so apathetic or even antipathetic toward teams? Part of the reason is insecurity, as indicated in Chapter 19. Another part is a conspicuously low consciousness



of teams in upper management. The team phenomenon, as we've described it, is something that happens only at the bottom of the hierarchy. For all the talk about "management teams," there really is no such thing—certainly never jelled teams at the managerial level. When managers are bonded into teams, it's only because they serve dual roles: manager on the one hand and group member on the other. They become accepted as part-time peers by the people they manage. As you go higher and higher in the organization chart, the concept of jelled teams recedes further and further into oblivion.

## Once More Over the Same Depressing Ground

Most organizations don't set out consciously to kill teams. They just act that way.



## Chapter 21

### A SPAGHETTI DINNER

Picture yourself a technical worker who's just been assigned to a new project. You know the manager and most of the other project personnel by name, but that's about it. Your first day on the new project is next Monday. On Wednesday before that Monday, you get a call from your boss-to-be. She's having a get-together, she says, for people on the new project. Is there any chance you could come by her place on Thursday evening for dinner with the rest of the team? You're free and want to meet the new group, so you accept.

When you arrive, the whole group is sitting around the living room drinking beer and telling war stories. You join in and tell a few of your own. The client liaison, who has also been invited, does a bit about his department head. Everybody has another beer. You begin to wonder about food. There is no smell of anything cooking and no sign of anyone working in the kitchen. Finally your boss-to-be admits that she hasn't had time to make dinner, and suggests that the whole crew walk over to a nearby supermarket and assemble the makings of a meal. "I guess we must be capable of putting a spaghetti dinner together."

#### Team Effects Beginning to Happen

Off you go. In the supermarket, you amble as a group through the aisles. Nobody takes charge. Your boss seems to have anything on her mind but dinner. She chats and laughs and offers up a story about the IRS. In spite of a general lack of direction, some things



do get thrown into the cart. One fellow has already gotten the salad pretty well taken care of. There is some talk of making a clam sauce, and when nobody's opposed, two of your new mates begin to talk out the details. You decide to make your patented garlic bread. Someone else picks out a bottle of Chianti. Finally there is a consensus that enough stuff is in the cart for dinner.

Back at the ranch, you all set down your bags of groceries and the boss grabs another beer and tells about a new software tool. Little by little the party gravitates toward the kitchen where some preparations are beginning. Your boss gives no direction, but she pitches in to chop onions when someone suggests that's what's needed. You start the garlic and olive oil simmering in a pan. There is a sauce bubbling and some spaghetti boiling. Gradually a dinner comes together. You all eat till you're full and then share in the cleanup chores.

### What's Been Going on Here?

So far, nobody has billed a single day of effort to the project, but you've just had your first success as a group. Success breeds success, and productive harmony breeds more productive harmony. Your chances of jelling into a meaningful team are enhanced by your very first experience together.

Presented this way, the spaghetti dinner may seem like a contrivance on the manager's part. But it probably wasn't and wouldn't have seemed like it had you been there. If you had asked the manager in question what she had in mind for the evening, she would have probably replied in total sincerity, "Dinner." A natural manager has got a subconscious feel for what's good for the team. This feel may govern decisions throughout the project. The entire experience is organized for small, easy joint successes. You have to look twice to see the manager's hand in any of this, it just seems to be happening.

Variations on the story of the spaghetti dinner have been told to us in different forms and about different managers for years. The common thread is that good managers provide frequent easy opportunities for the team to succeed together. The opportunities may be tiny pilot sub-projects, or demonstrations, or simulations, anything that gets the team quickly into the habit of succeeding together. The best success is the one in which there is no evident management, in which the team works as a genial aggregation of peers. The best boss is the one who can manage this over and over again without the



team members knowing they've been "managed." These bosses are viewed by their peers as just lucky. Everything seems to break right for them. They get a fired-up team of people, the project comes together quickly, and everyone stays enthusiastic through the end. These managers never break into a sweat. It looks so easy that no one can believe they are managing at all.