

Wicked Problems and Social Complexity¹

by Jeff Conklin, Ph.D.

Some problems are so complex that you have to be highly intelligent and well informed just to be undecided about them.

--Laurence J. Peter

This book is about collective intelligence: the creativity and resourcefulness that a group or team can bring to a collaborative problem. Collective intelligence is a natural property of socially shared cognition, a natural enabler of collaboration. But there are also natural forces that challenge collective intelligence, forces that doom projects and make collaboration difficult or impossible. These are *forces of fragmentation*.

The concept of fragmentation provides a name and an image for a phenomenon that pulls apart something which is potentially whole. Fragmentation suggests a condition in which the people involved see themselves as more separate than united, and in which information and knowledge are chaotic and scattered. The fragmented pieces are, in essence, the perspectives, understandings, and intentions of the collaborators. Fragmentation, for example, is when the stakeholders in a project are all convinced that their version of the problem is correct. Fragmentation can be hidden, as when stakeholders don't even realize that there are incompatible tacit assumptions about the problem, and each believes that his or her understandings are complete and shared by all.

The antidote to fragmentation is shared understanding and shared commitment. This book is about a new way to create shared understanding, and this chapter sets the stage by

¹ This paper is Chapter 1 in the book *Dialogue Mapping: Building Shared Understanding of Wicked Problems*, John Wiley & Sons. For more information see the CogNexus Institute website at <http://cognexus.org>.

exploring specific ways that the forces of fragmentation work in organizations and projects.

Fragmentation = wickedness **D**
social complexity



Fragmentation and Organizational Pain

There is a subtle but pervasive kind of pain in our organizations. It is characterized by such frequently heard complaints as “How am I supposed to get my work done with all of these meetings?” and “We always have time to do it over again, but never time to do it right.” It is a sense of futility of expecting things to be one way and repeatedly banging into a different reality. It is the dull ache of déjà-vu when you are handed an impossible deadline or a vague assignment. It is the frustration of calling a meeting to make a decision and watching the meeting unravel into a battle between rival departments, or get lost in a thicket of confusion over the meaning of a technical term. It is the frustration of finally achieving a hard-won decision and then having it fall apart or get “pocket vetoed” because there wasn’t really buy-in. It is the pain of fragmentation.

I was working late one evening when the janitor came in to vacuum the office. I noticed that he was going back and forth over the same areas without appearing to get the lint up off the carpet. I smiled and shouted to him (the vacuum cleaner was a loud one) “It must be frustrating to have to use that vacuum cleaner.” He looked at me with a sad smile and said “Not as frustrating as being told to go back and do it over!” It is that kind of pain, and it goes all the way up to the executive suite.

Part of the pain is a misunderstanding of the nature of the problems at hand. More precisely, the pain is caused by working on a special class of problems – *wicked problems* – with thinking, tools, and methods that are useful only for simpler (“tame”) problems. Problem wickedness is a force of fragmentation. Most projects today have a significant wicked component. Wicked problems are so commonplace that the chaos and futility that

usually attend them are accepted as inevitable. Failing to recognize the “wicked dynamics” in problems, we persist in applying inappropriate methods and tools to them.

Another force of fragmentation is *social complexity*, the number and diversity of players who are involved in a project. The more parties involved in a collaboration, the more socially complex. The more different those parties are, the more diverse, the more socially complex. The fragmenting force of social complexity can make effective communication very difficult. Social complexity requires new understandings, processes, and tools that are attuned to the fundamentally *social* and *conversational* nature of work.

For example, in a joint project involving several companies and government agencies, there was a prolonged struggle over the mission statement simply because of a terminology difference: each sponsoring agency had their own term for the core concept², and to pick one term meant disenfranchising one of the agencies. This is a very simple example of fragmentation of meaning.

Social complexity means that a project team works in a social network, a network of controllers and influencers including individual stakeholders, other project teams, and other organizations. These relationships, whether they are with direct stakeholders or those more peripherally involved, must be included in the project. For it is not whether the project team comes up with the right answer, but whose buy-in they have that really matters. To put it more starkly, without being included in the thinking and decision-making process, members of the social network may seek to undermine or even sabotage the project if their needs are not considered. Social complexity can be understood and used effectively, but it can be ignored only at great peril.

My janitor friend had an advantage over the rest of us in the organization because he could clearly see that his vacuum cleaner was not actually picking up the dirt. When we are working on wicked problems in a socially complex environment, it is much harder to notice that our tools are simply not “picking up the dirt.”

As we enter the new millennium the forces of fragmentation appear to be increasing, and the increasing intensity of these forces causes more and more projects to flounder and fail. The bigger they are, the more intense the fragmenting forces, the more likely the projects are to fail.

Moreover, the situation is not that project teams are aware of fragmentation and are taking appropriate measures to deal with it – quite the opposite, most teams accept fragmentation as inevitable. Indeed, most people are unaware of some basic facts about novel and complex problems. Managers, in particular, seem to be unaware that linear processes are not effective with such problems.

² The project concerned “unmanned aerial vehicles” (UAVs), also known as “remotely piloted aircraft” (RPAs).

Opportunity Driven Problem Solving

A study in the 1980's at the Microelectronics and Computer Technology Corporation (MCC) looked into how people solve problems. The study focused on design, but the results apply to virtually any other kind of problem solving or decision-making activity – the kinds projects are fraught with.

A number of designers participated in an experiment in which the exercise was to design an elevator control system for an office building. All of the participants in the study were experienced and expert integrated circuit designers, but they had never worked on elevator systems before. Indeed, their only experience with elevator systems came from riding in elevators. Each participant was asked to think out loud while they worked on the problem. The sessions were videotaped and analyzed in great detail¹.

The analysis showed, not surprisingly, that these designers worked simultaneously on *understanding the problem* and *formulating a solution*. They exhibited two ways of trying to *understand the problem*:

- efforts to understand the requirements for the system (from a one page problem statement they were given at the beginning of the session); and
- mental simulations (e.g. “Let's see, I'm on the second floor and the elevator is on the third floor and I push the 'Up' button. That's going to create this situation....”).

On the solution side, their activities were classified into *high*, *medium*, and *low* levels of design, with high-level design being general ideas, and low being details at the implementation level. These levels are analogous to an architect's sketch, working drawings, and a detailed blueprint and materials list for a house.

Traditional thinking, cognitive studies, and the prevailing design methods all predicted that the best way to work on a problem like this was to follow an orderly and linear “top down” process, working from the problem to the solution. This logic is familiar to all of us. You begin by understanding the problem. This often includes gathering and analyzing “requirements” from customers or users. Once you have the problem specified and the requirements analyzed, you are ready to formulate a solution, and eventually to implement that solution. This is illustrated by the “waterfall” line in Figure 1.

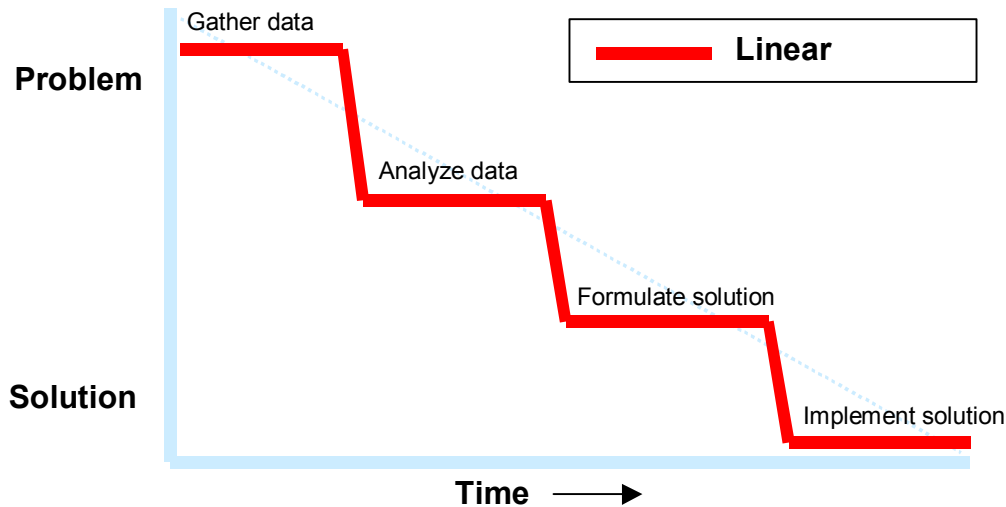


Figure 1: Traditional wisdom for solving complex problems: the “waterfall”

This is the pattern of thinking that everyone attempts to follow when they are faced with a problem, and it is widely understood that the more complex the problem is, the more important it is to follow this orderly flow. If you work in a large organization, you will recognize this linear pattern as being enshrined in policy manuals, textbooks, internal standards for project management, and even the most advanced tools and methods being used and taught in the organization. In the software industry it is known as the “Waterfall model,” because it suggests the image of a waterfall as the project “flows” down the steps towards completion.

However, the subjects in the elevator experiment did not follow a waterfall. They would start by trying to understand the problem, but they would immediately jump into formulating potential solutions. Then they would jump back up to refining their understanding of the problem. Rather than being orderly and linear, the line plotting the course of their thinking looks more like a seismograph for a major earthquake, as illustrated in Figure 2. We will refer to this jagged-line pattern as *opportunity-driven*², because in each moment the designers are seeking the best opportunity for progress toward a solution.

These designers are not being irrational. They are not poorly trained or inexperienced. Their thought process was something like: “Let’s see, idle elevators should return to the first floor, but then, you only need one elevator on the first floor, so the others could move to an even distribution among the floors. But the elevators need to be vacuumed regularly. I suppose we could add a switch that brought idle elevators down to the first floor. But then what happens in an emergency?” In other words, what is driving the flow of thought is some marvelous internal drive to make the most headway possible, regardless of where the headway happens, by making *opportunity-driven* leaps in the focus of attention. It is precisely because these expert designers are being creative and because they are learning rapidly that the trace of their thinking pattern is full of unpredictable leaps.

In particular, the experiment showed that, faced with a novel and complex problem, human beings do not simply start by gathering and analyzing data about the problem. Cognition does not naturally form a pure and abstract understanding of “the problem.” The subjects in the elevator experiment jumped immediately into thinking about what kind of processors to use in the elevator controller, and how to connect them, and how to deal with unexpected situations, such as if one processor failed. These are detailed solution elements.

These experienced designers illustrated that problem understanding can only come from creating possible solutions and considering how they might work. Indeed, the problem often can best be described in terms of solution elements. A requirement in the problem statement calling for “high reliability” was quickly translated into the idea of using a network of distributed processors – a high-level solution that drove the rest of the design process.

Figure 2 illustrates another striking observation: that problem understanding continues to evolve until the very end of the experiment. Even late in the experiments the designer subjects returned to problem understanding, the upper part of the graph. Our experience in observing individuals and groups working on design and planning problems is that, indeed, their understanding of the problem continues to evolve -- forever! Even well into the implementation of the design or plan, the understanding of the problem, the “real issue,” is changing and growing.

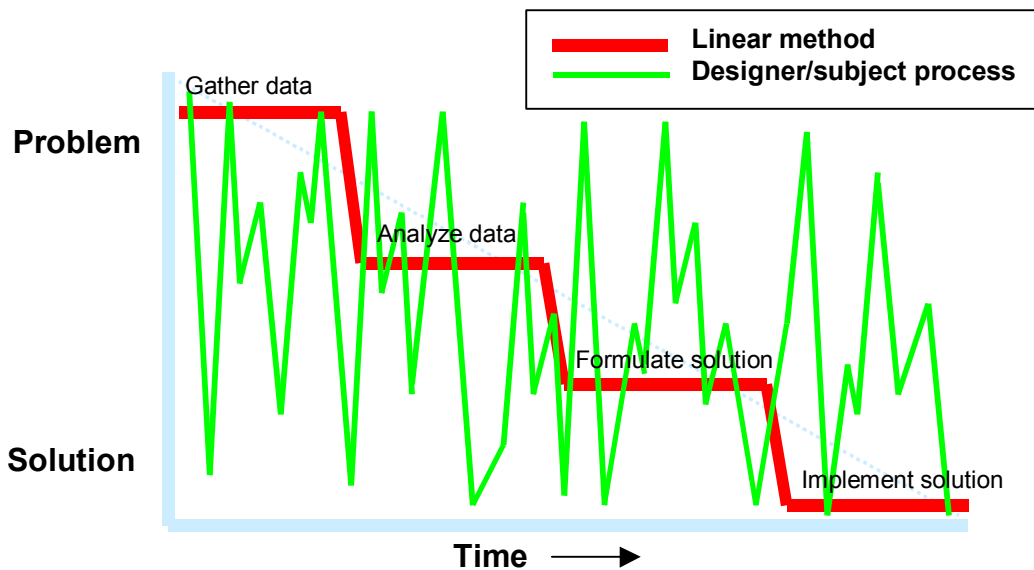


Figure 2: Pattern of cognitive activity of one designer -- the “jagged” line

The natural pattern of problem solving behavior may appear chaotic on the surface, but it is the chaos of an earthquake or the breaking of an ocean wave – it reflects a deeper order in the cognitive process. The non-linear pattern of activity that expert designers go through gives us fresh insight into what is happening when we are working on a complex and novel problem. It reveals that the feeling that we are “wandering all over” is not a

mark of stupidity or lack of training. This non-linear process is not a defect, but rather the mark of an intelligent and creative learning process.

In fact, this non-linear pattern does not come as a surprise to most people. Anyone who has ever worked on a complex project has the intuition that this jagged line process is what is really going on. But the experiment is significant because it gives us a real picture of the process that people follow when they really think about novel problems, and it is not the orderly and linear process we have been taught is proper!

From another perspective, the jagged line of opportunity-driven problem solving is a picture of *learning*. The more novel the problem, the more the problem solving process involves learning about the *problem domain*. In this sense the waterfall is a picture of already knowing – you already know about the problem and its domain, you know about the right process and tools to solve it, and you know what a solution will look like. As much as we might wish it were otherwise, most projects in the knowledge economy operate much more in the realm of learning than already knowing. You still have experts, but it's no longer possible for them to guide the project down the linear waterfall process. In the current business environment, problem solving and learning are tightly intertwined, and the flow of this learning process is opportunity-driven.

Some readers might object to this claim. Perhaps most folks in their organization have a strong sense of certainty about what is going on, a sense of confidence and pride in their knowledge of their business, and a sense that the problems the business is confronted with are quite manageable using the methodical application of well known rules and linear process logic. First, let me say, "Congratulations!" Certainly not all of the modern economy is knowledge based, and there are many who still enjoy a sense of quiet confidence and control in their professional lives. This book is not for them.

If your organization is a professional or consulting services business, or if there is a large technology (read "internet") component to your organization's products or business process, then you are all too familiar with the roller coaster ride of opportunity-driven problem solving. There are many reasons for this state of affairs, but one of the most important is that *you are operating in the realm of a special kind of problem: the wicked problem*. Wicked problems are one of the fragmenting forces mentioned at the beginning of this chapter, and it essential to understand the properties of wicked problems in order to counter and manage their fragmenting impact on projects.

Wicked Problems

The man who coined the term "wicked problem," Horst Rittel, was also the inventor of the Issue-Based Information System (IBIS) structure upon which Dialog Mapping is based³. Rittel and his colleagues perceived the limitations of the linear "systems approach" of design and planning over 30 years ago, and their research provides a foundation for what Rittel termed a "second generation" of systems analysis methodology. Rittel invented IBIS because, as an urban planner and designer, he found traditional

planning methods inadequate for the ill-structured problems he encountered in city planning.

Rittel's genius shines especially bright when we consider his solution for wicked problems: IBIS, a structure for rational *dialogue* among a set of diverse stakeholders. This is a perspective that puts human relationships and social interactions at the center, a perspective that is only now coming into vogue as a key insight of post-modern thought.

As Rittel defined them⁴, wicked problems are distinguished by the following characteristics:

1. *You don't understand the problem until you have developed a solution.*

Every solution that is offered exposes new aspects of the problem, requiring further adjustments of the potential solutions. Indeed, there is no definitive statement of "The Problem." The problem is ill structured, an evolving set of interlocking issues and constraints. Rittel said, "One cannot understand the problem with knowing about its context; one cannot meaningfully search for information without the orientation of a solution concept; one cannot first understand, then solve." Moreover, what "the Problem" is depends on who you ask – different stakeholders have different views about what the problem is and what constitutes an acceptable solution.⁵

2. *Wicked problems have no stopping rule.*

Since there is no definitive "The Problem", there is also no definitive "The Solution." The problem solving process ends when you run out of resources, such as time, money, or energy, not when some optimal or "final and correct" solution emerges. Herb Simon, Nobel laureate in economics, called this "satisficing" -- stopping when you have a solution that is "good enough" (Simon 1969)

3. *Solutions to wicked problems are not right or wrong.*

They are simply "better," "worse," "good enough," or "not good enough." With wicked problems, the determination of solution quality is not objective and cannot be derived from following a formula. Solutions are assessed in a social context in which "many parties are equally equipped, interested, and/or entitled to judge [them]," and these judgements are likely to vary widely and depend on the stakeholder's independent values and goals.

4. *Every wicked problem is essentially unique and novel.*

There are so many factors and conditions, all embedded in a dynamic social context, that no two wicked problems are alike, and the solutions to them will always be custom designed and fitted. Rittel: "The condition in a city constructing a subway may look similar to the conditions in San Francisco, say, ... but differences in commuter habits or residential patterns may far outweigh similarities in subway layout, downtown layout, and the rest." Over time one acquires wisdom and experience about the approach to wicked problems, but one is always a beginner in the specifics of a new wicked problem.

5. *Every solution to a wicked problem is a "one-shot operation."*

Every attempt has consequences. As Rittel says, “One cannot build a freeway to see how it works.” This is the “Catch 22” about wicked problems: you can’t learn about the problem without trying solutions, but every solution you try is expensive and has lasting unintended consequences which are likely to spawn new wicked problems.

6. *Wicked problems have no given alternative solutions.*

There may be no solutions, or there may be a host of potential solutions that are devised, and another host that are never even thought of. Thus, it is a matter of *creativity* to devise potential solutions, and a matter of *judgement* to determine which are valid, which should be pursued and implemented.

These criteria are more descriptive than definitional. The point is not so much to be able to determine if a given problem is wicked or not as to have a sense of what contributes to the “wickedness” of a problem.

Here are a few examples of wicked problems:

- Whether to route the highway through our city or around it?
- How to deal with crime and violence in our schools?
- What to do when oil resources run out?
- What should our mission statement be?
- What features should be in our new product?

While many of the problems that we will look at in this chapter are problems that occur in organizations, the above list should make it clear that many of the social problems that we face in our communities are also “wicked problems”.

Wicked Problem Example: A New Car Design

Let's consider a potentially wicked problem in the design of a new car. Let's imagine a project team that has formed around a new assignment: the Marketing department is asking for a design that emphasizes side-impact safety – they want to promote a new “safe car” to compete with Volvo. That is the problem to be solved, that is the work of the project. There is a deadline and a budget and a senior executive that the project reports to.

Now consider the criteria for a wicked problem again:

1. *You don't understand the problem until you have developed a solution.*

One approach to making a safer car would be to add structural support in the doors to make the car safer from side impact. It turns out that the additional door structure doubles the cost of the door, makes the doors heavier and harder to open and close,

changes the fuel mileage and ride, and requires an adjustment to the suspension and braking systems. Making the doors stronger leads into other design problems, but also bounces back into marketing problems such as “What should the price be?”, “How much do people really care about side impact survivability?”, “What do customers really want in a car?” All of these problems interact with each other. And at the senior executive level, the real question is “Should we continue this project to produce this new car?”

2. *Wicked problems have no stopping rule.*

When does the car become “safe”? There is no natural stopping point in working out the tradeoffs among safety, performance, appearance, and cost. At some point, the design team will be forced to make a decision. If it were not for project deadlines, the team would swirl indefinitely in “analysis paralysis”.

3. *Solutions to wicked problems are not right or wrong.*

No amount of study, laboratory experiments, or market surveys will establish that that project team’s solution is “correct.” Ironically, when the car gets produced, there will be reviews pointing out that the doors are heavy and difficult to open when parking on a hill, mixed with law suits from people who were injured in side-impact accidents despite the stronger doors.

4. *Every wicked problem is essentially unique and novel.*

Even if the project team has several successful car designs under its belt, the “safe door” problem is essentially unique and novel, because of the configuration of issues and stakeholders. First, a recent study by a consumer safety organization suggests that side impact injuries would be reduced by side air bags, which are not a part of the design. Second, a side-impact injury lawsuit has been filed against the company – if the new design is announced now, it may look like an acknowledgement of prior unsafe designs. Moreover, federal legislation is emerging that may put legal constraints on the strength of the doors. The design of safer doors is not merely a technical problem: it is a political and PR problem as well.

5. *Every solution to a wicked problem is a “one-shot operation”.*

The creation of a safer car is a one-shot operation. When the new safer car finally reaches the market, it may be a flop, or it may change the safety standards for the whole industry. The design team can build prototypes of the car and test them, but there is no way to anticipate the unintended consequences of producing and selling the new vehicle.

6. *Wicked problems have no given alternative solutions.*

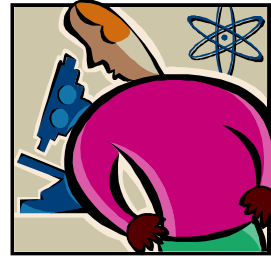
The safe door problem does not have a few discrete possible solutions from which to choose. There is an immense space of options in terms of structural reinforcement, materials, cushioning, window design, hinge placement, and how the door latches and opens. The design team cannot select from a few options – it must collectively exercise creativity and judgement about an elegant resolution of all of the design priorities.

The design of a new “safe car” is an example of a wicked problem. It cannot be solved by engineers alone, nor is there any way of determining that any given solution is “correct” or even optimal. It all depends on where you stand.

Coping with Wicked Problems

Not all problems are wicked. In contrast, a “*tame problem*” is one for which the traditional linear process is sufficient to produce a workable solution in an acceptable time frame. A tame problem:

1. has a well-defined and stable problem statement,
2. has a definite stopping point, i.e. when the solution is reached,
3. has a solution which can be objectively evaluated as right or wrong,
4. belongs to a class of similar problems which are all solved in the same similar way,
5. has solutions which can be easily tried and abandoned,
6. comes with a limited set of alternative solutions.



Finding the square root of 7358 is a tame problem, as is finding the shortest route from A to B on a map. Repairing a computer, raising \$10,000, and selecting a new doctor when you move to a new city are all tame, if complex and difficult, problems. Tame does not mean simple – a tame problem can be very technically complex.

A problem doesn't have to possess all six characteristics in order to be wicked. Putting a man on the moon was a problem with a lot of wickedness, for example, but also with some tame elements. There were certainly some wicked sub-problems. But notice that the main problem statement, putting a man on the moon and returning him safely, did not change over time (criterion 1). There was a definite “stopping point” at which we could say we had solved that problem (criterion 2). And the solution could be clearly evaluated as having succeeded or failed (criterion 3). It may be convenient to describe a problem as wicked or tame, but it's not binary – most problems have degrees of wickedness.

You also can't tell from the outside if a problem is going to be wicked. Like the safe car design example, many problems appear tame on the surface, but are indeed wicked once you get into them.

There seems to be a natural inclination to see problems as tame, and to avoid the wicked ones. Who wants to take on a problem that, by definition, can't be solved!?

The first step in coping with a wicked problem is to recognize its nature. There is a tendency to treat all problems as tame, perhaps because tame problems are easier to solve,

reinforced by the lack of understanding about wicked problem dynamics and the tools and approach they require. There is a psychological dimension here – a shift from denial to acceptance.

The command and control paradigm of management reinforces blindness about the true nature of the problem. Inherent in this paradigm is the idea that a person in charge gives the solution (the right solution, the only solution) to other people, who are in charge of implementing it. To function in such a hierarchy often means to collude in systematic denial of the complex and ill-structured dynamics of wicked problems, a phenomenon dubbed “skilled incompetence” by Chris Argyris (e.g. Argyris and Schön 1996).

As a result, there are two common organizational coping mechanisms that are routinely applied to wicked problems: *studying* the problem, and *taming* it.

While **studying** a novel and complex problem is natural and important, it is an approach that will run out of gas quickly if the problem is wicked. Pure study amounts to procrastination, because little can be learned about a wicked problem by objective data gathering and analysis. Wicked problems demand an opportunity-driven approach; they require making decisions, doing experiments, launching pilot programs, testing prototypes, and so on. Study alone leads to more study, and results in the condition known as “analysis paralysis,” a Catch 22 in which we can’t take action until we have more information, but we can’t get more information until someone takes action. One corporation I worked with, struggling to decide between two very different strategic paths for the future, studied and discussed the two options for so long that, by the time they had decided and implemented their choice, the chosen option was no longer viable.

Taming a wicked problem is a very natural and common way of coping with it. Instead of dealing with the full wickedness of the problem, one simplifies it in various ways to make it more manageable – to make it solvable! There are (at least) six ways to tame wicked problems, corresponding to the six criteria for wickedness:

1. Lock down the problem definition. Develop a description of a related problem or a sub-problem that you *can* solve, and declare that to be the problem. Resist all efforts to expand or modify the problem definition. For example, if the problem is how to reduce violence in schools, you could focus on the much more tractable problem of how to install metal detectors in all school entrances. As another example, in the software field, one learns to “freeze the requirements” as a way to lock down the problem⁶.
2. Assert that the problem is solved. Since a wicked problem has no definitive solution, the whole point of attempting to tame it is so that a solution can be reached. Usually this step requires locking the problem down (see point 1), though it is possible to simply assert that the problem is “solved” without clarity about what the problem was. Such assertions, however, generally require considerable authority to appear successful, such as in an autocratic organization or a dictatorship. As an example illustrates, one way of dealing with a United Nations resolution demanding that you destroy all weapons of mass destruction in your country is to simply assert that you

have done so. It should be clear that this approach to taming a problem depends critically on the evidence that you offer that the problem is solved.

3. Specify objective parameters by which to measure the solution's success. This is the measurement approach. For example, to find out if we have solved the problem of school violence, we might count the number of deaths and injuries on school property – if this measure drops to zero, then we have solved the problem. This taming approach amounts to locking the problem down (point 1), however, because what is measured becomes, officially and by definition, the problem. Whatever is not measured is then free to absorb the real problem. With intense enough focus, we might reduce the number of violent incidents on the school grounds to zero ... problem solved! ... but overlook new problems that had been created, such as a sharp rise in violent incidents just off of the school grounds.
4. Cast the problem as “just like” a previous problem that has been solved. Ignore or filter out evidence that complicates the picture. Refer to the previous solution of the related problem: “It’s just like that problem. Just do the same thing again.” For example, there is a saying in military circles that “we always fight the last war,” meaning the tendency to assume that the enemy will behave as he did in the last war.
5. Give up on trying to get a good solution to the problem. Just follow orders, do your job, and try not to get in trouble. Maybe the organization will fix the serious problems in the current solution in a revised version or release next year.
6. Declare that there are just a few possible solutions, and focus on selecting from among these options. A specific way to do this is to frame the problem in “either/or” terms, e.g. “Should we attack Iraq or let the terrorists take over the world?”

Different people prefer different coping mechanisms – some would rather study the problem until they really understand it, others, impatient with sitting around, would rather tame the problem to something manageable and jump into action.

However, attempting to tame a wicked problem, while appealing in the short run, fails in the long run. The wicked problem simply reasserts itself, perhaps in a different guise, as if nothing had been done. Or, worse, sometimes the tame solution exacerbates the problem.

Social Complexity

At the beginning of the chapter we asserted that the two most intensely fragmenting forces impacting projects today are wicked problems and social complexity⁷. These forces tend to co-occur. Can a socially complex group have a tame problem? Probably so. Can an individual have a wicked problem? Certainly. Yet the concepts are distinct: while wickedness is a property of the problem/solution space and the cognitive dynamics of exploring that space, social complexity is a property of the social network that is engaging with the problem.

Social complexity is a function of the number and diversity of players who are involved in a project. The more parties involved in a collaboration, the more socially complex it is. The more different those parties are, the more socially complex.⁸

Projects and problem solving have always been social in nature. Project success has always depended on collaborative skills and collective intelligence. But in earlier times of greater social homogeneity, the collaborative skills picked up on the playground were sufficient. The rules of engagement of family dynamics held in project meetings, and hierarchical authority could always be used to sort out the hardest parts. Now, in the “knowledge workforce”, more democratic models of decision-making are being used. Also, women have a far stronger role, often playing leadership roles. Minorities and foreign nationals are almost always present on the team. The old assumption that “we all pretty much think and act the same way” just doesn’t hold any more. In addition, organizations are flattening, opening up, and moving toward increased workplace democracy. More disciplines, departments, and dogmas are represented on the typical project team.

The jagged line graph from the MCC elevator study can help us visualize the impact of social complexity on a project. Imagine adding a second designer, represented by the dotted orange line in Figure 3, to help solve the elevator design problem.

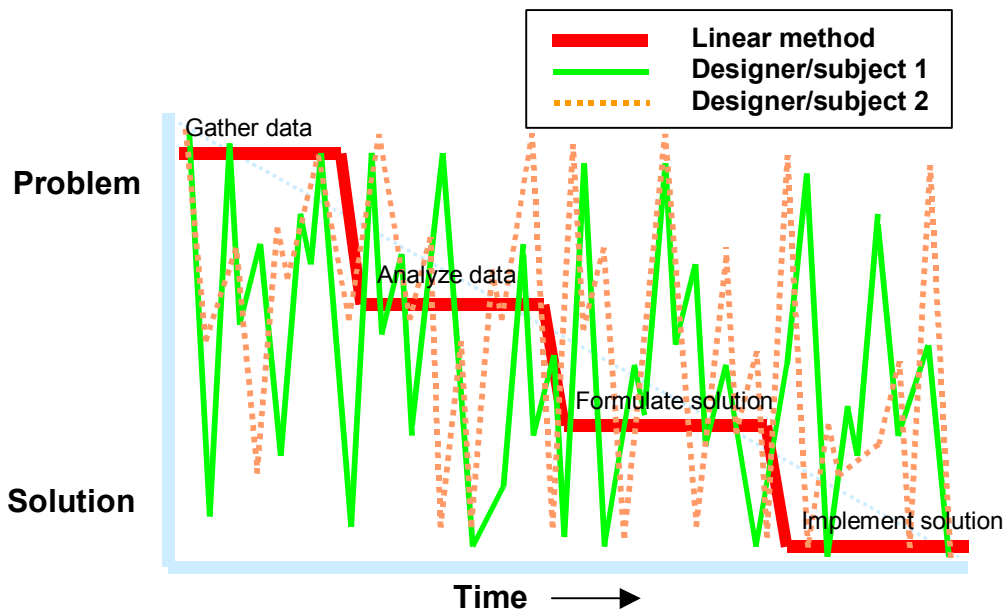


Figure 3: A wicked project with a second designer working on the problem

Notice that the second designer, like the first designer, goes through an opportunity-driven process between the problem and solution spaces, but the new designer’s thinking process is quite different in the particulars from the first designer’s. Since she has a different background and training, the pattern of her cognitive flow will also differ.

Let's imagine you are the project leader. You are the one who is responsible for the project being on time, in budget, and meeting all of its requirements. Even if you understand that the process is going to be opportunity-driven, you must still make plans, create schedules, allocate resources, and commit to milestones. You can't "plan" for the process to be opportunity-driven! Thus, in effect, you are officially in charge of keeping the project on the waterfall line.

Now let's consider two project team meetings, occurring at different points along the time line. At meeting A in Figure 4, you are a happy project leader because everyone is in synch, focused on the same activity, analyzing the requirements just like it says in "the book." Your prospects for bringing your project in on time and in budget look good.

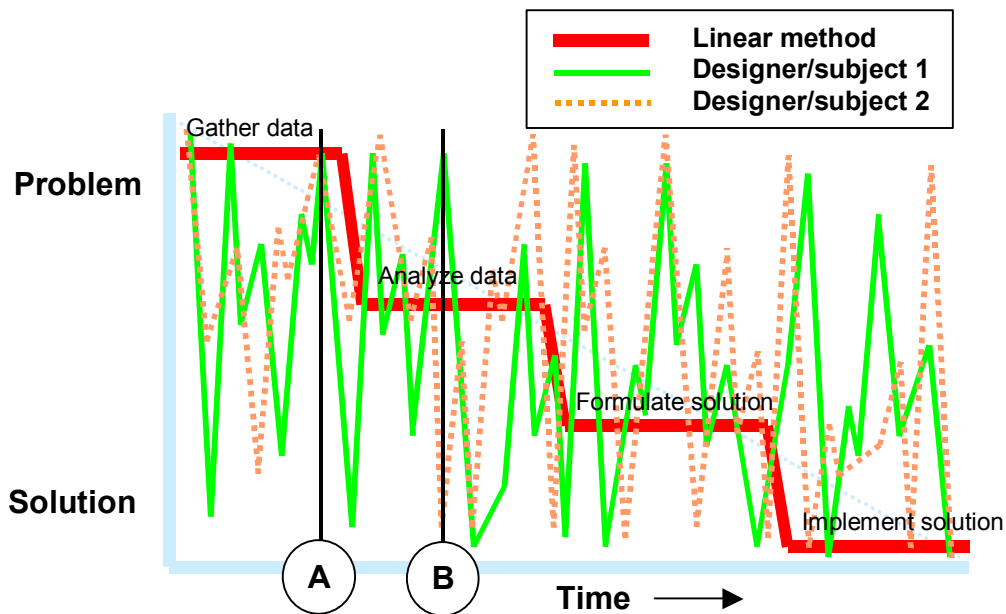


Figure 4: Two team meetings, A and B, during the project.

Some time later, at meeting B in Figure 4, the team has finished up with the data analysis and is now in the next phase, high-level design. But there are signs of trouble. Designer 2 looks tired but radiant. He says, "I was driving home last night and I had an idea. I stayed up all night programming, and ... you won't believe this, but ... I put together a program that does the whole thing. Sure, it still needs a little work, but, hey, we're practically done! Way ahead of schedule! I can't wait to show it to you!" In his personal opportunity-driven process, he has made a major leap, all the way to bottom of the chart, to the final solution.

There is a long pause. Designer 1 also looks tired, but not so radiant. Holding up the long-finished requirements document she says, "Sorry, we're not even *close* to done. I was with the client yesterday, and it turns out that there is a transaction that the system needs to handle that they never even told us about. They said it didn't have anything to do with our system. But it turns out it has a *lot* to do with our system. We've got to back to square one and start over!"

Neither of these key players is where you need them to be, according to the linear plan you created at the beginning of the project. You can feel chaos rising and control slipping away. You desperately plead with them to refocus on the high-level design, because, according to the calendar, that's where the project needs to be.

Perhaps you turn to the first designer and say something like "That's a good idea, Henry – but we really need to finish the high-level design. Can you hang on to that code for a while?" Turning to the other designer, you beg, "Look Sally, we already have gotten those requirements signed off. We can't go back. We'll just have to take care of that new transaction in the next release of the system."

This scenario exemplifies a tiny slice of the tension that is inherently part of working in a socially complex environment. Despite the most carefully thought out plans, wicked problem dynamics and the different jobs and orientations of the two fragment the project team and its process. The above scenario is mild – there are only three stakeholders involved in the project. As projects grow in size and organizations grow flatter, social complexity increases. Large projects typically have dozens of stakeholders, representing the project team, other departments, and other organizations. And not only does each one have their own "jagged line," they are likely to have different ideas about what the real issues are, and what the criteria for success are.

Consider the safe car design team. Bob, from Marketing, has been conducting studies and focus groups that indicate a lot of interest in cars that are safer in a collision. He is concerned with how to package a new "safe car" in a way that is positive, sexy, and up-beat. Christine, from Engineering, is very concerned about making the doors too heavy, but she has worked on structural integrity in the past and is excited about new technologies that, while expensive, could make the doors both stronger and lighter. Harry, the representative from the Management Team, sees the big issue as cost – top management is pushing affordability and value as the new strategy to increase sales. Alan, from IT, has a mandate from his management to get this team to use the new CAD (Computer Aided Design) system on this project. There are team members who represent Regulatory Affairs, Finance, Graphic Design, Power Train, and Quality Assurance, as well as team members from several major suppliers, including electronics and interior materials.

Each player has their own individual experience, personality type, and style of thinking and learning. Each player adds a new jagged line to the graph. The individual diversity among these players will make collective intelligence a challenge, and will make consensus virtually impossible to achieve.

But social complexity doesn't stop with individual diversity – each of these players comes from a different *discipline*, with its own specialized language and culture. When Bob is among his colleagues in Marketing, they share common knowledge, a common set of concerns, and shared ways of thinking about and dealing with those concerns. However, when Bob tries to talk to Christine, from Engineering, he finds that she has little knowledge of basic marketing concepts, and seems to be uninterested in them. It's as if

she were from a different country, speaking a different language. Thus, achieving shared meaning and shared context is especially difficult.

Moreover, social complexity goes beyond individual diversity and diversity among disciplines. The real corker is that these players represent different *organizations*. Each organization has its own function and charter, its own goals, and is managed by its own executive director. These organizations often have divergent goals. Marketing is trying to make its sales numbers, while Engineering is trying to win the Baldrige quality award. When the members of a project team come together to collaborate, they represent not only themselves but also their respective management chain in the hierarchy. Ideally, everyone in the organization is committed to the same thing, but operationally goals and agendas can be quite fragmented.

Thus, social complexity makes wicked problems even more wicked, raising the bar of collaborative success higher than ever.

For example, the main feature of a wicked problem is that you don't understand the problem until you have a solution. But with social complexity, "not understanding the problem" does not show up as innocent wonder about the mystery of the problem, nor does it usually occur as a thoughtful collective inquiry into the deeper nature of the problem.

Rather, "not understanding the problem" shows up as different stakeholders who are certain that *their* version of the problem is correct. In severe cases, such as many political situations, each stakeholder's position about what the problem is reflects the mission and objectives of the organization (or region) they represent. In such cases there is a fine line between collaboration and colluding with the enemy. How can you make headway on a mutually acceptable solution if the stakeholders cannot agree on what the problem is?

The answer to this question – and the Holy Grail of effective collaboration – is in *creating shared understanding about the problem, and shared commitment to the possible solutions*. Shared understanding does not mean we necessarily *agree* on the problem, although that is a good thing when it happens. Shared understanding means that the stakeholders understand each other's positions well enough to have intelligent dialogue about the different interpretations of the problem, and to exercise collective intelligence about how to solve it.

Because of social complexity, solving a wicked problem is fundamentally a social process. Having a few brilliant people or the latest project management technology is no longer sufficient.

This book offers a practical approach for creating shared understanding and shared commitment in a complex social network, and explores the underlying principles that make this approach so effective. But before we can get into the "solution" offered in this book,

there is a little bit more to understand about the collaborative challenge posed by wicked problems and social complexity.

The Polarity of Design

Most projects wrestle with large social networks and their attendant complexity. It would be a mistake, however, to think that small project teams can escape fragmentation. Design possesses a fundamental property that can make a team of *two* socially complex. All that is needed is a representative from each of the two polarities of design: what is needed (marketing), and what can be built (engineering).

Virtually all creative work is a process of *design*⁹. All problems call for *designing* a solution. All projects are essentially *designing* something. Design, in both the technical and artistic sense, is the process of creating something new – e.g., a new car, a strategic plan, a software program, a stickier web site, next year’s budget, a new environmental policy.

Any design problem is a problem of resolving tension between what is needed and what can be done. On the one hand, the process of design is driven by some desire or need – someone wants or needs something new. The need might be expressed by a customer, or it may be a guess about what the market wants. The need or want is expressed in the language of what *ought to be* – what should be done, what should be built, what should be written. On the other hand, the process of design is constrained by resources – what *can be done* given the available resources such as time and money and the constraints imposed by the environment and the laws of science.

Every need has a price tag – the process of design is about devising solutions that are feasible and cost effective. Going back to the safe car design, the need might be quite specific, e.g., the car must protect the occupants from harm if it is struck from the side by another vehicle of similar weight traveling 30 miles an hour. It may turn out that such a car would cost twice as much as a normally safe car. It may turn out to be impossible at *any* cost. Perhaps we have to change the need: reduce the required speed of safe impact to 10 miles an hour, because then it only increases the cost of the car by 15%.

Thus, in a very basic way, every project is about reconciling the fundamental polarity between the world of What-Is-Needed and the world of What-Can-Be-Built. These two worlds correspond to the upper and lower halves of the MCC elevator study diagram. In Figure 5, the upper half, being about understanding the problem, is focused out in the world on a specific client or user or market. There is always someone who has a need or a desire, and the task in the Problem or What-Is-Needed aspect of design is to specify that need. The lower half, being about the Solution, is focused “in the shop” on What-Can-Be-Built – what do we have the resources and skills and tools to actually make, and what will it cost and how long will it take.

As you can see, there is an immense difference between these two worlds. When an individual does design, she stands with one foot in each world. Moving back and forth between the two worlds, she tries to create a solution that joins the two polarities of design in an elegant way. Design teams have a bigger challenge. While it is possible for each person on a project team to be standing in both worlds, the tendency is for the polarity of design to be reflected in a polarity of roles. The world of What-Is-Needed is the domain of the Marketing and Sales department, and sometimes upper Management, whereas the world of What-Can-Be-Built is territory that belongs to the Engineering (or Manufacturing, Software Development, IT, etc.) department.

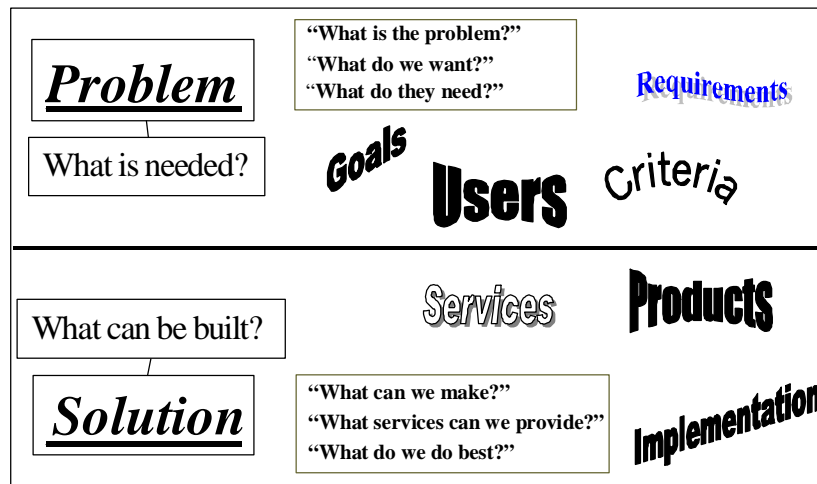


Figure 5: The two parts of the world of design

The inherent unity of the design process turns into a battle between departments. The world of What-Is-Needed, claimed by the Marketing department, becomes a self-referential world with its own culture and customs and language. The world of What-Can-Be-Built is claimed by the technologists, the nerds and hackers who actually build things, with its own culture and customs and language. When they sit down together on a project, the polarity of design turns into an inter-cultural war that is expensive, wasteful and ineffective, a war frequently featured in Scott Adams “Dilbert” cartoons.

Thus social complexity is not just a function of the *number* stakeholders – it is also a function of structural relationships among the stakeholders. While large projects have an increasing number and diversity of stakeholders, it only takes one player from each side of the polarity of design – one from Marketing and one from Engineering – to cause the collaborative gears to grind to a halt.

Technical Complexity

In addition to wicked problems and social complexity, technical complexity is a potentially fragmenting force. It includes the number of technologies that are involved in a project, the immense number of possible interactions among them, and the rate of technical change. For example, to be a serious player in the software industry today, your software must run on a variety of types of computers. Each type (or “platform”) has several operating systems, and each operating system has many versions that are currently in the field and must be supported. You must choose what language your software will be written in: Java, C, C++, Cobol, Fortran, etc. Each of these programming languages has a variety of supported versions (compilers); for example, Microsoft and Sun each have a major version of the popular Java language used in World Wide Web applications. Then you must choose the set of utilities (“library”) you will use for creating your user interface. There are dozens of other choices, and all of these options interact with each other. Moreover, the field is changing so fast that new options become available, and others drop into oblivion, almost every day.

As much as technical complexity raises the risk of project failure, it is also the most well recognized fragmenting force. So much has been written about technical complexity and how to deal with it, so many tools and methods are available, that there is little to add here. The Dialog Mapping approach presented in this book excels at dealing with complex technical information, but the real power of Dialog Mapping, and the point of this book, is to provide an approach and a set of tools for dealing with the non-technical side of fragmentation: wicked problem dynamics and social complexity.

Fragmentation and Coherence

We have described wicked problems, social complexity, and technical complexity as forces that fragment projects, causing them to fail. It is important to recognize that these forces are not due to incompetence, poor management, or any human failing. They are part of the “physics” of projects. There is no quick fix for the phenomenon of wicked problems. No glib formula about “Seven Steps to Crush Social Complexity” or “Tame Your Way to the Top”.

Moreover, the physics of fragmentation is obscured by a cultural condition of resignation, denial, and grim determination that has grown up around it. In my consulting and facilitation experience I have met this condition in organizations and on project teams. I have seen it manifest in many forms, sometimes as outright panic, sometimes as plodding determination, sometimes as a vague sense of futility. This condition of organizational pain is so chronic, however, that, like low-grade back pain, it has faded into the background of organizational experience and is taken for granted, assumed to be normal and inevitable. The condition is not wicked problems, nor social complexity – these are causes of the condition. Once this chronic condition is seen and understood, in my experience, a huge compassion emerges for what we are up against when we go to work.

For what we do accomplish and the courage that it takes. A whole new perspective about work and life opens up.

This is why it's so useful to distinguish the common element of fragmentation. Wicked problems fragment the process of project work, especially when the problem is misdiagnosed as tame. Wicked problems also fragment direction and mission – if you can't agree on what the problem is, how can you be aligned on a solution? Social complexity fragments team identity – the ideal of team unity is compromised by the dynamics of competing interests and hidden agendas. The duality of design tends to divide allegiances between requirements and implementation. Social complexity also fragments meaning – key terms and concepts are used in different ways by the different stakeholders. Project teams are often geographically distributed, further fragmenting relationships and communications. Participants in a modern project team are pulled in a thousand different directions by the centrifugal forces of wicked problems, social complexity, and technical complexity (see Figure 6).

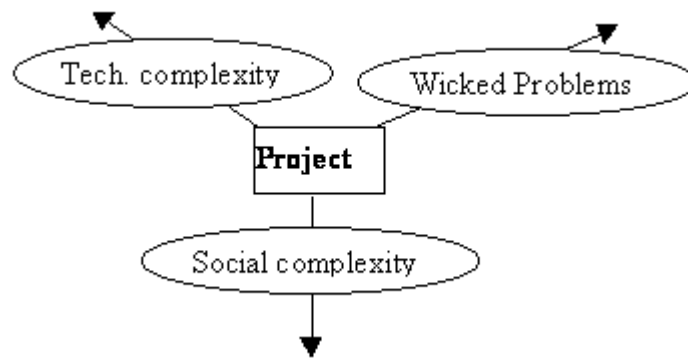


Figure 6: The "centrifugal" fragmenting forces pulling a project apart

The notion of fragmentation points to all of these problems, but it is pretty abstract. Because it points deep into the culture and practices of project work, it is difficult to observe fragmentation directly. There is, however, a more observable indicator of fragmentation: *blame*. Instead of seeing the systemic nature of project challenges and the value of social diversity, we tend to see a big mess, to view it as the result of incompetence, and to blame each other for it. We blame upper management for sending mixed signals or for lack of direction. We blame HR for poor hiring practices and lack of training. We blame the “bean counters” for over-tight budgets and lack of fiscal flexibility. We blame IT for confusion and the lack of stable infrastructure. We blame our customers for not knowing what they really want. We blame each other because we have different personalities and learning styles. (How many conversations do you notice in your organization that involve placing blame?)

In times of stress the natural human tendency is to find fault with someone else. We tend to take the problem personally, at an organizational level, and assume that

the chaos we see is a result of incompetence or, worse, insincere leadership. Since our education and experience have prepared us to see and solve tame problems, wicked problems sneak up on us and create chaos. Without understanding the cause, there is finger-pointing instead of learning.

Not so long ago most human illness was regarded as the result of evil spirits, so, when people got sick, the fix was to let the evil spirits out, by, for example, drilling holes in their heads. It wasn't very effective, but – within that system of thought – it was rational. These days, when big projects run into problems, we hold emergency meetings, then fire the consultants or rearrange the org chart. It isn't very effective, but it's a rational response to fragmentation ... *if* you believe that the problems result from human failing, i.e. poor performance or incompetence.

I was doing some training with a management team at a utility company several years ago. The Human Resources (HR) department had recently announced a new policy regarding the way employee performance would be evaluated and reported in the future, and these managers were very upset because the policy was so obviously flawed, and it had a direct impact on them. “What were they thinking?!?” and “Those morons in HR!” they exclaimed. As an exercise we decided to design a better policy. After an hour and a half we reviewed our solutions, and what do you suppose they realized? That it was a very hard problem, given the organizational and legal constraints in the system, and that, all things considered, HR had come up with a pretty good approach! They shifted from blame to deeper understanding of the problem.

If we step back and take a systemic view, we can see that the issue is not whose fault the mess is – the issue is our collective failure to recognise the recurring and inevitable dynamics of the mess. If we take a systemic view, we turn away from blame and away from easy technical fixes, and to look in the social domain – in building capacity to collaborate effectively on wicked problems.

As Rittel said, “We are now sensitized to the waves of repercussions generated by a problem-solving decision directed to any one node in the network, and we are no longer surprised to find it inducing problems of greater severity at some other node.”¹⁰

The antidote for fragmentation is coherence. How, then, do we create coherence? In organizations and project teams – in situations where collaboration is the life blood of success – coherence amounts to shared understanding and shared commitment. Shared understanding of meaning and context, and of the dimensions and issues in the problem. Shared commitment to the processes of project work and to the emergent solution matrix.

Coherence means that stakeholders have shared meaning for key terms and concepts, that they are clear about their role in the effort, that together they have a shared understanding

of the background for the project and what the issues are, and that they have a shared commitment to how the project will reach its objectives and achieve success. Coherence means that the project team understands and is aligned with the goals of the project and how to reach them. Coherence means that a wicked problem is recognized as such, and appropriate tools and processes are constantly used to “defragment” the project. With increased coherence, more collective intelligence becomes available to deal with change and complexity. Coherence means that despite social complexity there is a sense of ability and confidence in crafting shared understanding and negotiating shared meaning.

Summary

This chapter has been about laying a foundation that identifies the “problem” which Dialog Mapping addresses. This problem is:

- The powerful fragmenting forces of wicked problems, social complexity, and technical complexity;
- The confusion, chaos, and blame created by failing to distinguish these forces;
- The lack of tools and techniques for “defragmenting” project dynamics.

The process of Dialog Mapping is a powerful approach for addressing the problem of fragmentation, as it allows a diverse group of people to generate coherence around wicked problems. This group coherence is a necessary step toward addressing fragmentation, yet it is neither a silver bullet nor a cure-all. Given the complex nature of organizations, it is not sufficient for a single team or even multiple teams to achieve coherence; the organization as a whole needs to become a knowledge organization, and gain a kind of “literacy” or “fluency” in the language of coherence: distinctions, tools, methods, and practices for crafting shared understanding and shared commitment. Dialog Mapping is a first step toward that kind of literacy.

In the next chapter we explore the physics of coherence, and in particular the marvelously cohering forces of *shared understanding* and *shared commitment*. Following that, in Chapter 3, we shift gears and peek in on a Dialogue Mapping session.

Acknowledgements

Many people have contributed to this paper over its years of evolution. I am immensely grateful to Horst Rittel for his groundbreaking work in this area—this paper is dedicated to him. William Weil co-authored earlier versions of this paper, and his contribution still shines through in the current version. Kim Salins has been a stalwart ally for the past few years of evolution of this work. I’m also grateful for constructive reviews from Bill Pardee, Rosa Zubizarreta, Jack Park, and Eugene Eric Kim.

References

- Argyris, Chris, and Donald Schön, *Organizational Learning II: Theory, Method, and Practice*, Addison-Wesley, Reading, Massachusetts.
- Guindon, Raymonde (1990) Designing the Design Process: Exploiting Opportunistic Thoughts. *Human-Computer Interaction*, Vol. 5, pp. 305-344.
- Kunz, Werner, and Horst Rittel (1970) "Issues as Elements of Information Systems," Working Paper 131, The Institute of Urban and Regional Development, University of California, Berkeley, California, Tel: (510) 642-4874, email: iurd@uclink.berkeley.edu.
- Rittel, Horst (1969) "Reflections on the Scientific and Political Significance of Decision Theory," Working Paper 115, The Institute of Urban and Regional Development, University of California, Berkeley, California, Tel: (510) 642-4874, email: iurd@uclink.berkeley.edu.
- Rittel, Horst (1972) "On the Planning Crisis: Systems Analysis of the 'First and Second Generations'," *Bedriftsøkonomen*, Nr. 8. Also Reprint No. 107, The Institute of Urban and Regional Development, University of California, Berkeley, California, Tel: (510) 642-4874, email: iurd@uclink.berkeley.edu.
- Rittel, Horst (1972) "Structure and Usefulness of Planning Information Systems," *Bedriftsøkonomen*, Nr. 8. Also Reprint No. 108, The Institute of Urban and Regional Development, University of California, Berkeley, California, Tel: (510) 642-4874, email: iurd@uclink.berkeley.edu.
- Rittel, Horst and Melvin Webber (1973) "Dilemmas in a General Theory of Planning," *Policy Sciences* 4, Elsevier Scientific Publishing, Amsterdam, pp. 155-159. Also Reprint No. 86, The Institute of Urban and Regional Development, University of California, Berkeley, California, Tel: (510) 642-4874, email: iurd@uclink.berkeley.edu.
- Rittel, Horst, and Douglas Noble (1989) "Issue-Based Information Systems for Design," Working Paper 492, The Institute of Urban and Regional Development, University of California, Berkeley, California, Tel: (510) 642-4874, email: iurd@uclink.berkeley.edu.
- Simon, Herbert A. (1969) *The Sciences of the Artificial*, Second Edition, MIT Press, Cambridge, Mass.

¹ See (Guindon, 1990).

² Guindon, in the original paper, called it "opportunistic".

³ See the bibliography for references to Rittel's papers on wicked problems.

⁴ Actually, Rittel had a more exhaustive list of criteria for wicked problems. I have attempted to simplify the concept somewhat without losing its essence.

⁵ Wicked problems pose a terminology dilemma. There is no "the problem" in the traditional sense – like Heisenberg's elementary particles, getting close enough to the problem to see it ... changes it. Similarly, no "solution" is ever achieved, in the traditional sense. We might better use terms like "domain of concerns and needs" for "problem", and "domain of resolution and satisfaction" for "solution." However, for expediency and clarity, we will use the more familiar terms. So when we say "... developed a solution" we don't mean a real, final, mutually-acceptable solution, we simply mean a proposal that might resolve some part or aspect of a wicked problem.

⁶ Indeed, in most government contracts taming the problem has been refined to a high art: the problem definition, as described in the requirements document, becomes a contract which cannot legally be changed.

⁷ The term social complexity has been used in more academic contexts as well, in ethology and anthropology. For example, “Society is a web of dynamic interactions and interrelationships of individuals and their natural and artificial (human-made) worlds. The nature of this web, its self-organisation, evolution and transformation are at the focus of the study of social complexity. This study is centred in the rich conceptual basis of non-linear science the science of turbulence and chaos, emergence and fractals, self-organisation and criticality: the science of complexity.” From an unpublished paper “Social Complexity” by Vladimir Dimitrov (<http://www.uws.edu.au/vip/dimitrov/>).

⁸ See, for example, “Learning and Leading through the Badlands”, by David Berdish (<http://www.pegasus.com/levpoints/badlands.html>); “The complexity that Ford and other large organizations grapple with comes in two "flavors": system complexity and social complexity. System complexity derives from the infrastructure of the company—the business model it uses, the way the company organizes its various functions and processes, the selection of products and services it offers. Social complexity comes from the different outlooks of the many people associated with Ford—workers, customers, families, and other stakeholders from every single country and culture that Ford operates in.”

⁹ “*Design: the approach that engineering (and some other) disciplines use to specify how to create or do something. A successful design must satisfy a (perhaps informal) “functional specification” (do what it was designed to do); conforms to the limitations of the target medium (it is possible to implement); meets implicit or explicit requirements on performance and resource usage (it is efficient enough). A design may also have to satisfy restrictions on the design process itself, such as its length or cost, or the tools available for doing the design. In the software life-cycle (waterfall), design follows requirements analysis and is followed by implementation.*” [“Object-Oriented Analysis and Design with Applications”, 2nd ed., Grady Booch].

¹⁰ (Horst and Webber, 1973) See also “Disturbing Complexity” chapter of *Surfing the Edge of Chaos*, esp. the Lapp and reindeer story, p. 161, and the coyote control story, p. 151.