The Art of Business Value

Mark Schwartz, 2016 (https://www.amazon.com/Art-Business-Value-Mark-Schwartz/dp/1942788045)

“Business value isn’t something that is well understood by *the business*. … There is a legacy way of thinking that distinguishes between *the business* and the technologists. The business figures out what is valuable, puts it into a set of requirements, and tosses it over the wall to IT. IT then makes a commitment to cost and schedule and delivers. This made sense in a Waterfall world.”

Value means different things to different people, and to different types of companies (e.g., startup, non-profit, government).

Everyone is looking for the one true metric that defines value for work done within an org. Common metrics are net present value (NPV), return on investment (ROI), and market value added (MVA). Many of these neglect the Agile dimension of time.

Bureaucracy in its own right is not evil or wasteful. It arises because you need rules to work within systems. We’d do well to not let the processes calcify.

Value is everyone's responsibility -- not just the business, not just the product owner, not just the delivery teams, not just IT. Cross-functional teams (i.e., X-teams) with an experimental mindset and culture are most productive here.

Foreword and Preface (The Art of Business Value)

Quotations I highlighted…

- What was especially striking, though, was that the obstacles and the waste were being injected by some of the most intelligent people I’d ever worked with, and certainly the most committed and well-meaning.
- It didn’t make much difference to us in a Waterfall world – we cared about schedule and cost milestones. But in Agile practice, we only care about the delivery of business value. Which means we care about . . . what? That was the train of thought that led to this book. The more I explored the topic, the more critical it seemed to become. It seemed to have implications in how Agile teams fit into the enterprise, how we measure their success, how we go about causing cultural change, how we think about the IT function in a company, how we deal with compliance and bureaucracy, and how we choose and work with a product owner or on-site customer. It was about whether the way we practice Agility aligns with the philosophy behind it. It was about how I should do my job as a CIO.
- In the meantime, the cutting edge of Agile practice today – DevOps and Continuous Delivery – seems to be moving us toward smaller and smaller batch sizes of requirements, perhaps approaching single-piece flow.
- Mostly they say that the CIO should grab “a seat at the table” (that’s the executive table, where the grown-ups sit).
- Every decision we make in a software development project is ultimately a decision about business value. Feature trade-offs are decisions about business value. Risk management is about business value. Communication with the enterprise is about business value. Developer morale is about business value – it can affect the company’s cost in hiring and retaining developers, and it can affect the inclination of developers to innovate new value-creating solutions. Agile thinking is explicitly about business value: instead of delivering to schedule milestones, we deliver simply in the way that maximizes business value.
- If you an Agile practitioner – a developer, a tester, or Scrum master – then you face decisions of business value every day. You need to be able to speak the language of business value to communicate with the organization at large.

Ch 1: The Problem

Summary

- “The problem: business value, critical but elusive, remains at large. Our first set of clues leads nowhere.”
- Depending on which Agile source you read, *value* can mean delivery of business value or customer value.
- “I would like to suggest that the conflation of business value, customer value, and user value is outdated and well out of step with Agile practice.”
- “There can be many reasons why *business* objectives differ from *user* objectives.”
- Any Agile practitioner can benefit by understanding value; this isn’t just for finance folks.
- Typically anything that brings revenue has a cost to obtain that revenue.
- ROI is difficult to isolate, because some stories/features have intertwined values (e.g., what’s the value of the left-front wheel of a car). Also, depending on who’s getting what type of value, there are trade-offs.

Quotations I Highlighted

- DevOps, in a sense, is about setting up a value delivery factory – a streamlined, waste-free pipeline through which value can be delivered to the business with a predictably fast cycle time.
- Product-focused companies earn their revenues by delivering value to customers, it's true – but is *that* value the same as what we mean by business value?
- Features that deliver customer value do not necessarily lead to increased revenues, or they can be more expensive than the revenue they drive.
- And if the success of an Agile project is to be determined by the value it delivers, then we have to think of that value in terms of *outcomes* , not completed stories, and measure it as such. Releasing code is not the same as delivering business value; to know that we have delivered business value, we must both understand what business value is and be attentive to its outcomes.
- This might sound like an academic exercise: business value probably sounds about as interesting to Agile practitioners as bookkeeping and accounting – things that MBAs, people inclined to that sort of stuff, study in business school. I assure you that this is a mistake. … I will try to show that many of the difficulties we face in adopting and improving software development practices in an organization can be traced to business value and its interpretation.
- According to Mike Cohn, one of the clearest and most prolific writers on Agile practice, "the product owner is responsible for making sure the product earns a good return on the investment made in it."
- The fact that ROI has a name, an acronym, and sometimes a formula makes it sound reassuringly precise. We are probably aware that the product owner is not actually calculating an ROI metric for each user story, but we feel that the standard is at least approximately being applied.
- It's curious, once again, that ROI is not defined or explained, though we are told that Agile practice is all about maximizing it.
- Are we going in circles, defining business value in terms of ROI, which is then defined in terms of business value?
- The difficulty is that the "return" in the equation can be pretty much anything. … But making investment choices based on a function of profit, as we will see, can lead to poor decisions.
- Why not simply use sales, or revenues? Because we are building a set of features that customers value, shouldn't we measure value by the sales that result? For one thing, focusing only on revenue would ignore any costs that the new features bring to the business.
- The first problem is that profit does not consider the *timing* of the cash flows from sales and costs. … Second, with ROI we are not considering the *risk* of the expected returns (or the cost, for that matter). … Thirdly, *profit* is based on financial accounting reports and is not intended for managerial decision making.
- Even if ROI were a good proxy for business value, it would not be very useful to product owners for prioritization decisions. In "The Problems with Estimating Business Value," Mike Cohn points out that it is difficult to assign value to individual stories, because the values of user stories are often intertwined. As examples, he asks what the values are of the left front wheel of a car or the doors and windows of a house.
- Dean Leffingwell, who has written extensively on Agile requirements, notes that prioritizing features through ROI is challenging because it involves making trade-offs between different types of value, and revenues generally cannot be allocated on a feature-by-feature basis.
- What is the impact of profitability of a dashboard that enables management to drill down on sales by region? There undoubtedly is a connection, but assessing it involves so many assumptions that the exercise is impractical.

Ch 2: The Meaning

Summary

- Whether your a publicly traded company, private company, non-profit, or government entity, it's all a game where you get to pick what rules apply and which numbers to use when it comes to value.
- Perhaps some of the capitalist mantras are falling out of fashion: https://hbr.org/2020/09/companies-used-to-share-how-each-dollar-of-revenue-was-spent
- Non-profits often use a dual bottom line matrix to make decisions (one axis = low-to-high impact, second axis = low-to-high financial stability)
- Much of the value for government organizations is difficult to quantify. For example, what value do you place on security? That is, how much money would you invest to prevent a terrorist activity – $1M? $100M?
- Perhaps government representatives, who "represent the collective aspirations of the citizenry," ideally are non-profits, but their mandates may shift depending on political cycles. I would assume that most government agencies are about operational work (e.g., issuing driver licenses, maintaining roads) rather than delivering new product offerings.
- Michael Porter (from Harvard Business School) states that a company is more likely to be successful by picking one of the following strategies: *cost leadership* (cheaper than competitors), *differentiation* (something others don't have), or *focus* (particular market the competitors don't serve or don't serve well).
- Agile is built on the idea that the business cannot know all of the requirements in advance – they must be *discovered*. Likewise, "…business value must be discovered, must be learned, must be turned into a testable basis for valuing requirements."

Quotations I Highlighted

- Ultimately, in a capitalist economy the duty of a corporation is to return value to its owners. … Such an approach is referred to as "Management by Value" or "Shareholder Value Approach."

- I believe that the major lessons covered in an MBA program can be reduced to two principles: (1) There is a time value of money. (2) A business venture needs a sustainable competitive advantage. Principle one says that a business should generate cash flows, preferably as soon as possible, and principle two says that in order to continue to generate cash flows, it needs a way to continue competing effectively in its market.
- So the value of an investment clearly depends on both how long it will take to pay off and what alternatives you have for investing the money.
- If you think about valuing a company as a whole – it is, after all, a sort of machine for producing cash flows – its value depends on its ability to *sustain* its cash flows. And that, of course, depends on whether it has a sustainable competitive advantage.
- For example, an early-stage VC investor may be focused on ensuring that the company's next round of funding can be raised at a higher valuation – in other words, their biggest concern may be to ensure that the company is perceived as more valuable when it next tries to raise money, because that will cause less dilution to their ownership stake. (Geoff's note: There's no one recipe; this can look like whatever it needs to look like.)
- [For non-profits…] The criteria for its success – that is, its definition of business value – is about accomplishing the mission for which it was chartered.
- The idea that there is a single metric that represents or can serve as a proxy for business value is also misguided; in order to have a complete picture of business value, we must consider the goals of the particular organization, the interests of at least some of its stakeholders, and a variety of indicators of value, some of which may be quantifiable and some of which may not.

Ch 3: The Culture

Summary

- Book cited throughout the chapter: *X-Teams: How to Build Teams the Lead, Innovate, and Succeed* by Debora Ancona and Henrik Bresman
- Having a product owner role simplifies some interactions, but can decouple the team doing the work from understanding the business.
- There seems to be more value in having fully integrated teams, rather than integration points where one type of value is translated into another type of value.

Quotations I Highlighted

- …[Ken Schwaber and Jeff Sutherland, the founders of Scrum] go as far as to say that "the Development Team *isn't allowed* to act on what anyone else says." Scrum's product owner is the OPYCLT – the *Only Person You Can Listen To*.
- The main advantage of loose coupling in software design is that it hides the details of *implementation*: as long as the product owner (or interface) can be relied on, the organization does not need to know the details of what the Scrum team is doing, and the Scrum team does not need to know the details of how the organization is determining business value.
- "High performing teams manage across their boundaries, reaching out to find the information they need, understand the context in which they work, manage the politics and power struggles that surround any team initiative, get support for their ideas, and coordinate with the myriad of other groups that are key to a team's success." – Ancona/Bresman
- An X-team, a team that has multiple touch points with the organization and even influences the organization's understanding of what it *sh ould* value, might be more appropriate.
- The product owner is given control over business value trade-offs because he is an expert on the business needs; is it reasonable to expect that his expertise extends to security needs, code maintainability, and performance engineering?
- I believe that the product owner model is based on a hidden assumption: that both the functional requirement and the security requirement can be translated into ROI terms and compared apples-to-apples.
- If we really want the Agile team to be responsible for delivery of business value, we need to give the team ownership over business value discovery and interpretation, not just delivery on requirements.
- There is no need for a single interpreter, since the entire team can learn together. The findings of the team then influence how the remainder of the organization thinks about business value. This is the model that DevOps points toward: an inclusive model where the team has all the skills necessary to create value.
- Our goal is to deliver value, to figure out how to meet the needs that are determined by the organization, and yet we consider the organization to be the biggest impediment to doing so. The only explanation I can think of for this is that we are implicitly assuming that there is a stable, objective, preordained definition of business value, and we are determined to deliver on that definition *despite* the organization around us. In my experience, this arrogance is not warranted; in fact, the organization probably understands value in ways the Agile team does not, and the obstacles to Agile adoption actually tell us something useful about business value in the organization.
- We can take an Agile approach to Agile adoption, driving organizational change incrementally based on what we can learn about the organization's true needs and dynamics.
- "Culture is a pattern of shared tacit assumptions that was learned by a group as it solved its problems of external adaptation and internal integration that has worked well enough to be considered valid and, therefore, to be taught to new members as the correct way to perceive, think, and feel in relation to those problems." – Edgar H. Schein
- What we must do in adopting an Agile practice is to both change and absorb that culture and those processes through an inspect-and-adapt approach.
- But the DevOps spirit goes further, looking to eliminate the conflicting incentives of organizational silos and the inhuman behaviors that can result from conflicting incentives.

Ch 4: The Rules

Summary

- Bureaucratic behavior generally follows from a business need.
- There's a fine line between checklists and processes that *keep things consistent and mistakes from happening* and *systemic distrust of the workers*.
- The author takes aim at Mike Cohn (Agile), Mary Poppendieck (Lean), and others for stating that certain practices "aren't optional" or "not allowed."
- Minimum Viable Bureaucracy = rules that fit the current business needs

Quotations I Highlighted

- We have a tendency to solve problems by creating standard processes rather than by relying on human judgment.
- Recent trends in software development have moved us even closer to bureaucratopia: we now consider it part of the developer's job to write tests that "prove" that his or her code does what it's supposed to do. Doesn't that sound like the mounds of paper devoted to ensuring that the bureaucrats have followed the appropriate processes - the required signatures and routing sheets and checklists?
- The point that I will make here is that bureaucratic behavior generally follows from a business need, and the only way to remove or improve the bureaucracy is to find a way to meet the underlying need.
- The purpose of an Agile team is to self-organized and meet the *underlying* business need in the best way possible, often by cutting through the bureaucracy.
- [Edgar] Schein also points out that a normal part of organizational growth is the creation of rules and paperwork to make up for the loss of face-to-face contact.
- When bureaucracy entered politics, it was seen as an improvement on arbitrariness, capriciousness, nepotism, and other undesirable characteristics of public administration.

Characteristic	Advantage
Division of labor	Efficiency through specialization
Managerial hierarchy	Clear chain of command
Formal selection	Based on merit and expertise
Career orientation	Freedom from external pressures
Formal controls	Efficiency
Impersonality	Protection from arbitrariness

- Of course, [Max] Weber and other authors are aware of the problems with bureaucracies. Rules become set in stone and can't change with circumstances. Rigidity discourages innovation. Rules themselves come to seem arbitrary and capricious. Their original purpose gets lost and the rules become goals rather than instruments. Bureaucracies can become demoralizing for their employees.
- My point is not about whether bureaucracy is good or bad. It is rather about the real business needs that are served through bureaucratic behavior.
- The US government is based on a system of "checks and balances" – in other words, a system of distrust. The great freedom enjoyed by the press, especially in reporting on the actions of the government, is another indication of the public's lack of trust in the government.
- Government agencies, therefore, place a business value on "optics" – how something appears to the observant public.
- Others, like the requirements of the Payment Card Industry (PCI), are imposed by non-government organizations. Some are imposed by auditors. Some are self-imposed. In theory, those requirements can be translated into Shareholder Value Added, Net Present Value, or Return on Investment because they affect the amount of risk of future cash flows. Meeting PCI requirements allows companies to accept credit cards, which probably increases revenues. Getting a clean audit builds trust in the company and affects its ability to raise capital, thereby supporting its market value.
- Similarly, John Stuart Mill considered bureaucracy just a form of administration that "accumulates experience, acquires well-tried and well-considered traditional maxims, and makes provision for appropriate practical knowledge in those who have actual conduct of affairs." In other words, it is simply a form of administration that captures knowledge in rules and makes it available to experts to apply.
- The pioneers and leaders of the Agile world have seen people make mistakes in exactly these areas, and they (rightly) know better. The rules are a form of institutional memory: the Agile community has learned what works, and these rules represent the learning.
- "Naturally companies try to institutionalize the lessons derived from their successes." [Hirotaka Takeuchi and Ikujiro Nonaka] However, they warn that "institutionalization, when carried too far, can create its own danger." That danger, is of course, is petrification, the result we often see with bureaucratic rules.
- Sure, we can change the rules, but we'd better make sure that the underlying need is still satisfied, because satisfying the underlying need delivers business value.

Ch 5: The CIO

Summary

- Many organizations treat IT as an order taker instead of a partner with the customer who makes value decisions.
- IT is treated with Waterfall-like behavior in an Agile environment, hence the problems that arise
- Having product owners have sole responsibility for defining value, especially in an IT org, doesn't make sense

Quotations I Highlighted

- Critical questions about how the Agile teams relate to the IT organization
    - How does IT leadership ensure the delivery of value and participate with the rest of the organization in defining value?
    - What role does an IT organization play when teams work directly with the business to create value, when teams self-organize to deliver value, or when a product owner is the only voice that expresses business value decisions?
    - If Agile frameworks and DevOps in particular are based on an inclusiveness that extends beyond the boundaries of IT, then what is it that IT is accountable for?
    - What can IT provide if not the functional silos of technical expertise that it has provided in the past?
- The legacy view considers the IT organization a service provider to *the business*.
- Some authors have described this role as "IT as an order taker." Notice how neatly this model fits in with the Waterfall approach to system delivery. The business chooses a set of requirements that it believes will add business value. It communicates them to the IT organization – on paper, of course. The IT organization responds with a plan and estimates of cost and schedule. The business agrees to the plan. The IT organization executes the plan and delivers a product to the business. The business accepts the product and then tries to derive business value from it. The model is the "negotiating a contract" model that the Agile manifesto de-emphasizes.
- You might say that in the classic Waterfall approach, IT is like the person behind the counter in a fast food restaurant, taking orders and then serving them up. With the shift to an Agile culture, IT became more like a salesman in a fashionable clothing store. Now IT helped the customer decide what he or she really wanted by making suggestions and giving the customer outfits to try on in order to elicit information about what the customer was really looking for.
- The one point all the authors seem to agree on is that the order-taker mentality just doesn't work.
- In his comprehensive study of how to apply Lean thinking across the entire enterprise, Jez Humble points out that the distinction between IT and the business is an outdated one.
- In other words, business value decisions rest within IT as well as *the business*, and the IT manager is in much the same position that we earlier associated with product owners: lacking a simple, single way to determine value.
- But it remains difficult to see how the autonomous, encapsulated Agile team can participate in this store of knowledge, expertise, and guidance if it only listens to the product owner on questions of business value. Does the prioritization of security-related stories necessarily fall to the product owner? Or is it part of a CIO's vision of how to manage risk across the enterprise? What if the CIO believes that moving from a private datacenter to the public cloud is essential for the long-term IT strategy, and that move requires some work from one of the product teams? Is this for the product owner to decide? What if it has no net value for this particular project, but it does for the organization as a whole? The product owner may be incentivized to disregard or de-prioritize a valuable unit of work.
- To overcome the limitations of the order-taker model, the IT organization must become an interpreter of business value as well as a provider of technical skills, and its interpretation must influence not only the behavior of project teams but also the enterprise's business strategy.

Ch 6: The Clue

Summary

- Many of the "classic" approaches – Waterfall, net present value (NPV), return on investment (ROI), and market value added (MVA) – work best when you have a static situation or when all but a few of the variables are known.
- Many point-in-time estimates are impractical to compute (e.g., how much value will this feature bring).
- Perhaps the author gets into this in a later section, but the advantage of having separate roles (e.g., dev, product owner, product manager) is to abstract responsibilities. The skills become too diluted when each person must know everything soup-to-nuts. Sure, you can have T-shaped people, but you still need the "stem" to go deep, and that falls on each person on the team for a specific skill.
- "Business value is a hypothesis held by the organization's leadership as to what will best accomplish the organization's ultimate goals or desired outcomes."
- "…Agile lets us put off decisions until the last responsible moment and to change decisions over time."

Quotations I Highlighted

- …business value decisions are strongly influenced by uncertainty about the future…
- The Waterfall model was based on taking a point-in-time snapshot of the information we know and using it to create a long-term plan that we would adhere to.
- In a world where technology is so deeply embedded in business, and in an environment of team empowerment, this distinction [business vs. development team responsible for determine value] seems troublesome to me.

- How is the product owner as OPYCLT [only person you can listen to] or as *driver* different from a manager in a hierarchical organization, with the team reporting to her? Perhaps you'll say that the product owner just tells the team what to do, not how to do it. But this is precisely what a good manager *does* anyway.
- Is it possible that the product owner role feels like a good idea because it helps fit Agile practice into a command-and-control hierarchy?
- How can the team "own" value delivery if their responsibility extends only to delivering features that they neither choose nor oversee the adoption of?
- A team that "owns" business value delivery must advocate for its own ideas and negotiate its boundaries. It needs skills of persuasion, communication, and emotional intelligence – sales skills. Why shouldn't it also have strategic skills? A team of generalizing specialists – T-shaped people – with IT savvy and the ability to build things themselves. Isn't this a powerful model in a digital services world?
- An important function of leadership in a CAS [Complex Adaptive System] is to provide context and incentives that nudge the system toward the desired outcomes.
- Context, coordination, cultivation, creating conditions for success – all of these suggest a role for management that is about influencing the evolution of the organization in a direction that delivers business value.
- …leadership's hypothesis is not a single plan of activities, but rather a set of business value indicators that will add up to the desired goal.
- Our Agile approaches, however, are concerned with managing risk by taking actions over time, rather than by accepting it as given.
- As we will see in the next chapter, evidence shows that a very large number of ideas do not wind up creating value; people are not very good at predicting how much value will flow from a new idea. Calculating NPV, ROI, or other point-in-time value metrics by having a person predict the cash flows will result from a new idea is unlikely to lead to good decisions.
- …Scenario Planning increases our agility because it keeps all scenarios alive as possibilities, and it trains us to recognize which scenario is occurring and to react accordingly.
- Each feature we build has value not only for its direct cash flows, but for the increase in the organization's agility: its ability to change course or build on the feature as appropriate in the future.

Ch 7: The Delivery

Summary

Although the author doesn't provide a magic recipe, he outlines eight "plays" to influence how you think about business value

- **Build the pipeline** – you need continuous delivery to build/measure/learn
- **Extend the pipeline** – don't just deliver; understand the business impact at the team level
- **Search for gems in the waste** – limiting yourself only to *customer* or *user* value is myopic
- **Explore the fourth dimension** – (time) attach value to keeping options open until you have more information
- **Polish the hairball** – enterprise architecture becomes a hairball over time; learn to manage it by treating corporate IT as an asset
- **Govern wisely** – aim for single-piece flow (small batches)
- **Change partners** – consider having the product owner not be the only person you can listen to (OPYCLT)
- **Feed the CIO** – this isn't just about managing IT, but about the org's use of technology

Quotations I Highlighted

- Prescriptions seem especially inappropriate in a book that promotes experimentation and makes the case that even the very definition of business value varies from organization to organization.
- The first need when adopting a business value-centric Agile approach is to set up a production process that allows for rapid, informative, and reliable feedback cycles.
- The pipeline is an automated bureaucracy: it applies its rules in a rigorous, unemotional way, *sine ira et studio*. [without anger and passion]
- The testing process plays a special role in this DevOps bureaucracy. Tests are simply rules in version control.
- But the [Continuous Delivery] pipeline is a learning bureaucracy. It is tuned through retrospectives and continuous improvement; at any given moment it embodies the participants' beliefs about the best way to generate value through software delivery.
- There is also a new feedback loop. At a larger scale, the enterprise has a hypothesis about what creates ultimate value – shareholder value, mission value, owner satisfaction, or whatever it might be.
- As I pointed out in chapters 3 and 4, some of this "waste" is actually adding value. In a government agency with responsibility to the public and to Congress, proving that you are following the rules and using public resources effectively is actually a value-adding activity.
- Co-opt compliance folks by bringing them in early and making them part of the team.
- Building feature A may have value in itself, but it may also have value because it facilitates building feature B. A flexible architecture has extra value because it facilitates creating future value in a cost-effective manner.
- Like it or not, all of the Agile teams are adding to the [Enterprise Architecture] hairball. This seems overlooked in the Agile literature: we think of autonomous-ish teams building products defined by product owners, but we rarely reference the impact on the enterprise hairball and its implications. New features produced by teams are the sticky stuff – chewing gum, duct tape, wax, technical debt – that get attached bit by bit to the hairball.
- Douglas Hubbard provides an excellent way to think through this value and make decisions on whether to invest in obtaining information. He describes information as a way to reduce risk when making a decision: the value of the information is how much risk is removed, or

more technically the reduction in the Expected Loss if the decision turns out to be wrong. Hubbard's approach is perhaps too limited, since it only considers the value of information used to support decisions. It does not directly address, for example, the value of information that might persuade an investor to invest in your company, or the value a government agency is being able to report results to Congress.

- We learn on the scale of single requirements, but make investment decisions on the scale of programs or investment themes – thus the impedance mismatch.
- This is part of a broader question about how the Agile team relates to the entire enterprise of which it is a part. Is it "loosely coupled" through a product owner or other business representatives? Does it fit into an organizational hierarchy – say, an IT department? Is it "managed" somehow in a chain of command? Does the organization need to change in order to accommodate the Agile team?
- In *To Sell is Human*, Daniel Pink reports that his studies show that people who are in non-sales positions actually spend about 40 percent of their time at work engaged in sales activities: persuading, convincing, and influencing people.
- Think of the product owner or onsite customer as an information source, a team member, a helper – anything but an OPYCLT.
- …the CIO's influence extends throughout the entire organization through his management of IT assets, strategy, and value creation. The CIO applies information technology to help steer the entire organization.
- Heavy-handed command-and-control will generally not produce the best results, because it will not harness the creativity and diversity of the team. As a *strategic* approach, it fails. However, this does not mean that command-and-control *tactics* should never be used. Based on observations of how the teams are performing, management needs to decide on the appropriate tactical forces to apply.