MANNING

SAMPLE CHAPTER

Software Development Metrics

David Nicolette

FOREWORD BY George Dinwiddie

Software Development Metrics

by David Nicolette

Chapter 6

contents

6

Reporting outward and upward

This chapter covers

- Protecting the team from administrative details
- Reporting hours for capitalized projects
- Reporting project status for traditional, adaptive, and mixed projects
- Minimizing the effort of management reporting

At higher levels of management, people make different kinds of decisions than would be made at the team level. They need to know what's going on across an entire program, an entire product suite, an entire value stream, or the entire enterprise. Whereas you make decisions about how to keep the work moving forward, they make decisions about whether a project should be continued or canceled, and whether a given initiative ought to be capitalized or expensed. Details about individual work packages or user stories aren't helpful for that kind of decision-making. Those individuals need relevant, summarized information without any clutter.

At the same time, software development teams must be able to focus on their work without being distracted by administrative activities such as recording the number of hours they worked or estimating the expected completion date for a given software feature. Distractions like those interrupt flow, create delay, increase lead times, raise costs, and cause stress.

In this chapter, you'll see how to provide measurements that are useful to upper management without expending excessive effort in doing so. You'll also explore some of the issues that can arise when management expectations for upward reporting conflict with the metrics you've chosen to help you steer your work at the team level. You'll also see how you, as the person responsible for reporting metrics, can collect the data you need without interfering with your teams' day-to-day work.

I suggest that one of your functions is to act as an insulating layer between your teams and the organization. What does this mean? Others in the organization need to collect information from teams. Yet every interruption and every administrative task reduces the teams' effectiveness. As a person responsible for tracking progress, you can help teams maximize their effectiveness by handling administrative requirements on their behalf.

6.1 Reporting hours

Few issues cause more grumbling on technical teams than the requirement that people track the number of hours they spend working on each project. Contemporary wisdom in software development is to dedicate each team to a single project at a time. Even then, team members don't enjoy taking time to track the hours they spend on project work compared with the hours they spend on other tasks. In traditional organizations, teams may work on multiple projects concurrently, and in many organizations individuals may be allocated to multiple teams and multiple projects in a complicated matrix structure. Tracking the hours spent on each project each day can become a time-consuming activity whose value isn't obvious to technical workers.

Management needs to know where the time is spent for capitalized projects. Capitalization provides a tax benefit by allowing companies to spread the initial development cost of a software solution over the anticipated production lifetime of the solution. Management has to track costs closely. All the costs except labor are pretty straightforward to track, because they're mostly fixed costs. To get the labor cost, management depends on staff to report the hours they spend working on each capitalized project.

This is probably uninteresting to you, and it's definitely uninteresting to your teams, because your focus is execution, not funding. Knowing how each team member spends each hour doesn't help you execute the project. Spending time tracking hours certainly doesn't help team members complete any work. It isn't a value-add activity; it's administrative overhead. Pursuant to your function as an insulating layer, you need to provide this information to management without asking anything of your team members.

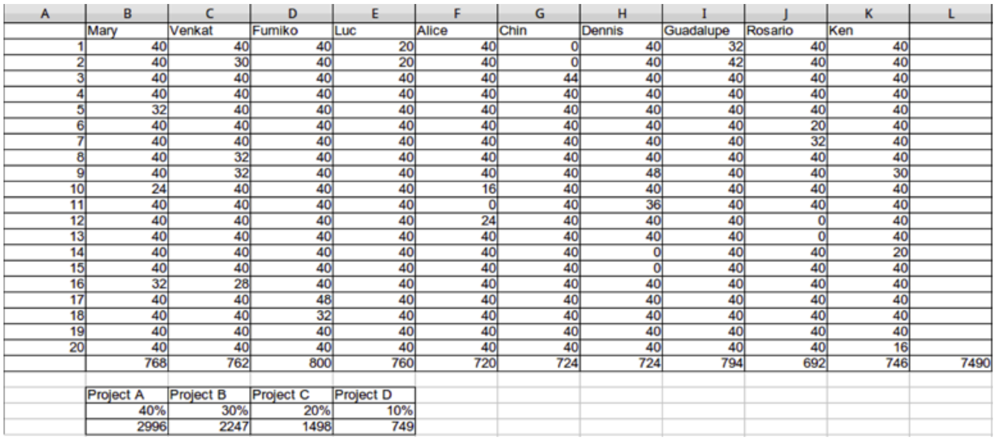
But if you don't ask them, how will you know how team members spend their time? The answer is clear, if you remember that you want to provide decision-makers with information *at the right level of detail.* Management doesn't need to know exactly how Mary Smith spent every minute of every day. They need to know that Mary's team spent more or less the expected amount of time working on capitalized project A and on capitalized project B.

6.1.1 An example

Let's say that over the span of projects A and B, the team is expected to spend about half their time working on each. Management needs to know whether that expectation was met. They don't need to know that in week 1 the team split their time 70/30 between the two, and in week 2 they split their time 30/70. That level of detail doesn't help management make the decisions they're responsible for making. They certainly don't need to know all the minor variations of each individual team member's time; they only need aggregate numbers.

To report accurate information about hours at the right level of detail, you can depend on some basic assumptions. First, you can assume that variations in time allocation will wash out over the course of a project. Approximate data will be accurate and useful to support management decision-making. Second, you can assume that most individuals will work a standard week unless something unusual happens. You needn't ask people to report their hours in painstaking detail—you only need to know about variances from the standard work week. Figure 6.1 shows an example of how this can work.

The spreadsheet tracks the total work hours put in by each of 10 team members on a team that's working on four projects. The team is allocated 40% to project A, 30% to project B, 20% to project C, and 10% to project D. Most of the time, team members work a 40-hour work week. You don't need to ask them how many hours they work in a normal week—it's standard. When they're on vacation, it's on the team calendar, and you don't need to ask them. When they're out of the office, they let you know by



A	B	C	D	E	F	G	H	I	J	K	L
	Mary	Venkat	Fumiko	Luc	Alice	Chin	Dennis	Guadalupe	Rosario	Ken	
1	40	40	40	20	40	0	40	32	40	40	
2	40	30	40	20	40	0	40	42	40	40	
3	40	40	40	40	40	44	40	40	40	40	
4	40	40	40	40	40	40	40	40	40	40	
5	32	40	40	40	40	40	40	40	40	40	
6	40	40	40	40	40	40	40	40	20	40	
7	40	40	40	40	40	40	40	40	32	40	
8	40	32	40	40	40	40	40	40	40	40	
9	40	32	40	40	40	40	48	40	40	30	
10	24	40	40	40	16	40	40	40	40	40	
11	40	40	40	40	0	40	36	40	40	40	
12	40	40	40	40	24	40	40	40	0	40	
13	40	40	40	40	40	40	40	40	0	40	
14	40	40	40	40	40	40	0	40	40	20	
15	40	40	40	40	40	40	0	40	40	40	
16	32	28	40	40	40	40	40	40	40	40	
17	40	40	48	40	40	40	40	40	40	40	
18	40	40	32	40	40	40	40	40	40	40	
19	40	40	40	40	40	40	40	40	40	40	
20	40	40	40	40	40	40	40	40	40	16	
	768	762	800	760	720	724	724	794	692	746	7490
	Project A	Project B	Project C	Project D							
	40%	30%	20%	10%							
	2996	2247	1498	749							

Figure 6.1 Hours applied to capitalized projects

phone or email, and you don't need to ask them. You never need to interrupt their focused work time to ask them to track their hours.

What percentage of their time did each team member devote to projects A, B, C, and D? It makes no difference. They're professionals, and they will spend whatever time is necessary to get their work done. For purposes of upward reporting, you can assume that they spent approximately the correct amount of time on each project. All you need is their total work hours, and you can calculate the percentages to report for each project.

6.1.2 Aggregate numbers are approximate

You might worry that this approach will lead to inaccurate upward reporting of hours. What if the team actually spent 90% of their time on project D, but you're reporting 10%? That's theoretically possible, but in the grand scheme of things, it isn't significant. In real life, technical professionals aren't meticulous about recording their work hours in exactly the right buckets. It isn't the focus of their attention at work. It isn't what they were hired to do. They merely record hours in whichever buckets they're authorized to use, to satisfy the time-reporting system. In many cases, they're only able to enter the number of hours into each bucket that were preallocated to the projects; they couldn't report their true hours even if they wanted to. The approximate numbers you report are probably more accurate than the numbers the technical staff have been recording until now.

This is an example of how you can function as an insulating layer between teams and the organization. You're providing sufficiently accurate information to track labor costs for capitalized projects, you're providing it at the right level of detail to be useful to decision-makers, and you're doing it in a way that doesn't degrade the teams' delivery performance.

6.2 Reporting useless but mandated metrics

Preparing this short section has been a challenge, because it would be all too easy to say the wrong thing. Management books typically don't refer to metrics as "useless." Organizations don't typically consider anything they mandate to be useless. No one wants to be called out for requiring others to report useless numbers. No one wants to waste time reporting useless information. And yet it happens.

In addition to helping your team, you're probably required to report certain metrics outward and upward in your organization. Sometimes the required metrics aren't useful for tracking progress or for supporting process improvement. I'm not talking about metrics that the organization uses but that you don't need at the team level; those are useful, even if they aren't of direct interest to you in a team-level role. I'm talking about metrics that no one uses or *could* use. The numbers pile up on reports and presentation slides, and then they're set aside. Often, when you try to find out who uses the metrics and for what purpose, no one can tell you. Perhaps the metrics served a purpose at some time in the past. Things have changed, but the administrative requirement to report the metric remains in force. When this is the case, your *performance review* depends on reporting the required metrics, but your *actual performance* depends on measuring the things that matter and not wasting time on activities that don't add value.

As a person at the ground level with direct responsibility for delivery, how can you deal with the situation in a way that's both pragmatic and ethical? Many people face this question every day. Fortunately, there are tactical and strategic actions you can take. At the tactical level, your goal is to insulate your teams from the damaging effects of inappropriate measurement. At the strategic level, your goal is to help the organization adopt more appropriate metrics.

6.2.1 Categories of problematic metrics

You need certain measurements to help your teams deliver and improve. Your organization needs certain measurements rolled up so that management can manage programs and the organization. But sometimes the organization also requires you to roll up metrics that aren't useful to anyone. How can you ensure that you're using your time wisely? Figure 6.2 shows one way of categorizing metrics that you report outward and upward.

Metrics that are useful to *someone* have to be reported. The question is how much of your valuable time to dedicate to reporting them. Consider the lower-right quadrant in figure 6.2. This represents metrics that help you and your teams deliver. This is where you need to spend most of the time that you devote to measurement. These are the metrics you've chosen based on an analysis of the development approach, process model, and delivery mode your teams are using. You're using these metrics to steer the work and to inform process-improvement efforts.

Some of the metrics that are useful to you are also useful to others in the organization. This is represented by the upper-right quadrant in the illustration. These are a subset of the metrics you use at the team level; you don't need to report everything you

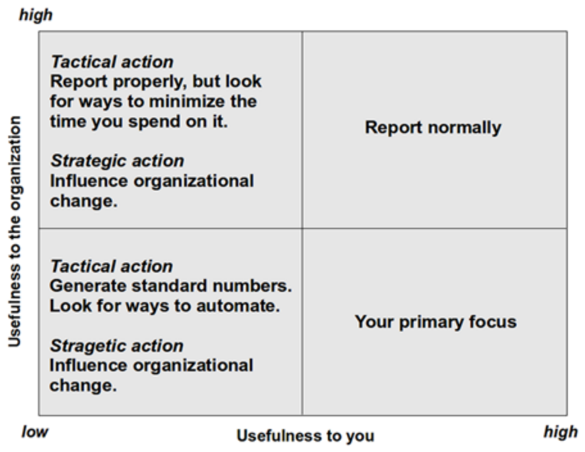

high
Tactical action
Report properly, but look for ways to minimize the time you spend on it.
Strategic action
Influence organizational change.
Report normally
Tactical action
Generate standard numbers. Look for ways to automate.
Stragetic action
Influence organizational change.
Your primary focus
Usefulness to the organization
low
Usefulness to you
high

Figure 6.2 Reporting metrics outward and upward

measure for the teams. By reporting these metrics, you're making a positive contribution to the organization. But your primary focus is delivery. Therefore, you want to look for ways to automate reporting. Many organizations implement project-tracking tools that can roll up metrics automatically. Take full advantage of such tools so that you can minimize the amount of time you spend reporting this information.

The left side of figure 6.2 represents reporting requirements that aren't of direct value to your teams. The items in the upper-left quadrant are important to the organization, even if they don't directly help teams deliver. Your tactical goal is to automate the production of these numbers while insulating the teams from the administrative burden of tracking the numbers. A strategic goal, should you choose to pursue it, is to influence the decision-makers in the organization to change the measurements they're using. Techniques for organizational change are beyond the scope of this book.

The lower-left quadrant represents metrics that help no one. Why would any organization require these? It happens when people adopt a new process or methodology "by the book," without thinking about which elements of the new process are a good fit for the organization. These metrics can drive unintended behaviors. It's important that you insulate teams from dealing with this sort of reporting. At best, it will distract them from their work; at worst, it will destroy their morale.

Because the metrics in this quadrant are, by definition, not useful to anyone, it follows that the numbers can be neither right nor wrong. People report these numbers only because they're required to do so. The people who consume the numbers make note of the fact that lower-level personnel have duly reported them, and then they file the numbers away. They aren't used. When you identify metrics like this, you can automate the generation of "standard" numbers for purposes of reporting. This can't do any harm, because the numbers aren't meaningful and are never used to support business decisions. They're merely an administrative requirement.

Why haven't I listed the metrics in this dangerous category? Because different metrics are useful in different situations. A metric that's helpful in one situation may be useless in another. I can't guess which metrics will fall into this category in your context. It's up to you to understand how the work flows in your organization and measure accordingly.

6.2.2 Recognizing what's really happening

The title of this section has been the theme of the book all along, but it bears repeating. It's common for organizations to officially adopt whatever metrics are defined for the formal delivery process they've chosen. In most cases, people overlay a defined process on the organization without giving much thought to adapting the process to the organization or adapting the organization to the process. And yet most of these defined processes assume that the organization will have the characteristics that enable the process to function as intended. There's a gap between theory and reality—between intention and action.

As each metric is described in the book, there's an indication of the conditions under which the metric can be useful. This is based on three key aspects of delivery:

- *Development approach*—When scope, schedule, and budget are all fixed at the outset, the approach is *traditional*. When one or two of those factors is flexible by design, the approach is *adaptive*. Metrics that are appropriate for the approach you're using will be useful, and others won't. This is true regardless of which formal process is ostensibly in use.
- *Process model*—The process you're using is almost certainly a hybrid. The good news is that it probably resembles one of the four reference models (linear, iterative, time-boxed, or continuous flow) more closely than the other three. Metrics that apply to the closest-matching process model will be useful, and others won't.
- *Delivery mode*—Some teams are executing on discrete projects that have a beginning and an end; others are supporting a product or technical infrastructure on an ongoing basis. Metrics that apply to the delivery mode that's in use will be useful, and others won't.

Most formal processes *assume* a particular approach, process model, and delivery mode. Many organizations *assume* that when they adopt a given process, the organization automatically conforms with the expectations of that process. In those cases, some of the metrics you're required to report outward and upward may not be meaningful, and they may even be counterproductive.

6.2.3 Beware of motivational side effects of metrics

In addition to the mechanical aspects of delivery, it's also necessary to be aware of unintended motivational effects of metrics. Let's consider one metric in particular, because it's widely used with popular frameworks for scaling agile methods in large organizations. When using a time-boxed process model, many organizations ask teams to commit to a fixed scope of work in each iteration. This notion of *commitment* is derived from an early version of Scrum, which has since been corrected. Scrum now calls for teams to *forecast* near-term future delivery performance based on their own velocity. Unfortunately, the word *commit* continues to be used, having been adopted in other processes besides Scrum, and is included in popular agile scaling frameworks.

The problematic metric is the *percentage of the iteration commitment that the team actually delivers*. Organizations that use this metric typically set a target for performance. For example, teams might be expected to deliver 80% of the amount of work they commit to deliver in each iteration. Proponents of the metric insist that it isn't intended as a target, but in real life team members perceive it to be a target they must hit, or else.

No doubt you can see the problems with this already. Velocity is a trailing indicator intended to be used for empirical planning. The moment you set a target for it, you drive undesired behaviors. A more insidious problem is that team members will assume that their individual performance reviews depend on hitting the target.

Commitment is a serious matter, after all. We commit to our families, our communities, our professions, our countries. Commitment is life or death. Commitment is what inspires us to make personal sacrifices for our children. Commitment is what leads us to jump on a grenade to save our comrades. When required to commit, teams will sacrifice evenings, weekends, holidays, vacations, family time, and health. At least, they will do so for a while. Ultimately they will game the numbers, and the metric will be useless for its intended purpose.

6.2.4 Understanding what the numbers mean

Let's consider another example that's a bit less damaging. Many organizations have adopted an iterative or time-boxed process with the expectation that it will instantly enable them to deliver faster. At the same time, they don't change their expectations with regard to planning, funding, and delivery. Most of these processes were designed for adaptive development, although they can be applied to traditional development as well. The problem is measuring something that isn't happening—trying to track traditional development using metrics for adaptive development.

The metrics defined for time-boxed processes are designed to be used with adaptive development. When a project is traditional (fixed scope, schedule, and budget), then the metrics don't mean what the literature says they should mean. That doesn't necessarily make them useless, but in your position you have to be aware of what information is really tracked, or you may overlook indications of emerging delivery risks.

Time-boxed processes can be used with traditional delivery as a way to break up work into small chunks that are easy to plan, estimate, and track. In this case, velocity doesn't reflect production-ready solution increments; it merely reflects completed work items, which may be interim artifacts such as unrealized requirements specifications, unexecuted test plans, undeployed software components, or untested code units.

This isn't necessarily damaging, provided you understand what the numbers mean. You'll probably use a burn chart to forecast delivery performance based on the pseudo-velocity observations. In this context, the burn chart is showing *percentage of scope complete to date*. As you know, both types of chart look like two lines: expected performance versus actual performance to date, with a trend line showing forecast future performance. So, you can label the chart *Burn Chart* and use it as a chart of percentage complete to date. In keeping with the traditional approach, you can use the chart to try to bring the project back on plan, rather than using it to adjust scope or schedule as you would do with adaptive development.

6.3 Summary

To maximize their effectiveness, teams need to focus on value-add work. Part of your function is to insulate them from the distractions of administrative activities while still providing information that others in the organization need from your teams.

It's necessary to report hours worked per project for capitalized projects so the organization can enjoy the tax benefits of amortizing the cost of new software solutions. Part of your function is to provide this information without interfering with your teams' focus on delivery.

Throughout the book, I've emphasized the importance of understanding how work flows in your organization so that you can select appropriate metrics for steering and process improvement. This chapter touched on the importance of *not* using metrics that *don't* align with the approach, process model, and delivery mode in your organization. Doing so can cause undesired behavioral effects that can be both subtle and damaging.

SOFTWARE ENGINEERING

Software Development Metrics

David Nicolette

When driving a car, you are less likely to speed, run out of gas, or suffer engine failure because of the measurements the car reports to you about its condition. Development teams, too, are less likely to fail if they are measuring the parameters that matter to the success of their projects. This book shows you how.

Software Development Metrics teaches you how to gather, analyze, and effectively use the metrics that define your organizational structure, process models, and development methods. The insights and examples in this book are based entirely on field experience. You'll learn practical techniques like building tools to track key metrics and developing data-based early warning systems. Along the way, you'll learn which metrics align with different development practices, including traditional and adaptive methods.

What's Inside

- Identify the most valuable metrics for your team and process
- Differentiate "improvement" from "change"
- Learn to interpret and apply the data you gather
- Common pitfalls and anti-patterns

No formal experience with developing or applying metrics is assumed.

Dave Nicolette is an organizational transformation consultant, team coach, and trainer. Dave is active in the agile and lean software communities.

To download their free eBook in PDF, ePub, and Kindle formats, owners of this book should visit manning.com/software-development-metrics

Free eBook
SEE INSERT

"A real boon to those making the transition from a traditional serial development model to an agile one."
—From the Foreword by George Dinwiddie

"Provides a solid foundation for how to start measuring your development teams."
—Christopher W. H. Davis
Nike, Inc.

"Measurement is the key to making and consistently hitting scheduling targets. This book will help you confidently build a schedule that is accurate and defensible."
—Shaun Lippy
Oracle Corporation

ISBN 13: 978-1-617291-35-7
ISBN 10: 1-617291-35-8
54799
9 781617 291357

MANNING $47.99 / Can $55.99 [INCLUDING eBOOK]