Sensible service management

A practical step-by-step guide to better IT service management

By Rob England, the IT Skeptic

CITRIX®

Table of contents

Practice 1: Better Process Can Be Better Business 4

Practice 2: Delivering to Your Customers 8

Practice 3: Responding to Requests 14

Practice 4: Response to Incidents 18

Practice 5: Dealing with Major Incidents 23

Practice 6: Problem Management 28

Practice 7: Change Management 31

Practice 8: Release Management 37

Practice 9: Deciding to Improve Service Performance 40

Practice 10: Measuring Service Improvements 45

Practice 11: Improving Service Performance 52

Sensible service management

11 no-nonsense, practical steps to better service management.

Often smaller businesses can struggle with the ideals of IT service management, as manifest in all the rules and guidelines called ITIL. This *Sensible service management* guide written by Rob England, the IT Skeptic, aims to cut through all the noise and jargon to tell IT managers why service management can be useful and why it doesn’t need to be so complex. The guide is broken into 11 best practices.

Practice 1: Better process can be better business

Small-to-medium enterprises (SMEs) take risks. All the time. Sure, in theory there are many things that ought to be managed and under control, but you simply don't have the bandwidth to deal with them. So you don't.

Indeed, experts who berate us about how we "should" do this and "must" do that remind me of an air force doctor advising World War II bomber pilots to give up smoking. But those pilots have got far more pressing risks to worry about. So, as an IT manager or IT consultant, you note the risks, fix a few, mitigate some, and live with the rest.

Instead you focus on what matters:

1. successful daily operations
2. growing the business
3. controlling costs
4. keeping customers happy

Probably in about that order.

When process wonks like me come along suggesting you improve your processes, you tend to lump us in the "give up smoking" category of advice. You say you don't have time to do process. You don't have the resources to investigate and implement the 1001 best procedures outlined in the ITIL guidelines.

But wait just a second.

There are ways to get some fast wins out of process improvement. There is a way to do it quick and dirty to make it serve your needs. That's what I want to talk about.

Now by "process," I mean standardized procedures or steps that use proven best practices to achieve your business goals. Once those procedures are put into place, you then measure and monitor them to see how well they work for you.

If we think about the four top goals I just mentioned, **process improvement** can help you deliver on these goals when you elevate processes in three areas:

1. how you respond to user requests
2. how you fix things
3. how you control changes to make sure you don't break things in the first place.

If you already know about service management (SM), you know what we are getting at here. If not, don't worry we'll break it down in a future section.

For right now, let’s focus on making it achievable for SMEs.

1. **Keep it simple**
   Process improvement doesn’t need to be the big deal that process consultants and books make it seem. If you open up one of the main SM references – ITIL – you get thousands of pages of “musts” and “shoulds.” But they are painting a picture of “best” practice: what the world would look like with infinite time and money. If we focus on the main needs and risks to meet our four goals, it all boils down to some simple concepts. In future posts, we’ll give you some simple models for what you need to think about.

2. **Look at what matters right here right now**
   You should pick and choose among those simple concepts so you can address what is burning your toes in your organization today. **If we start with where we are and work with what we have in order to deliver what we absolutely need most, then any process improvement becomes manageable.**

3. **Make sure it is worth it**
   If you are focused on successful daily operations, growing the business, controlling costs, and keeping customers happy, anything you do is likely to give a positive return on investment (ROI). In another blog post we’ll talk about simple methods for estimating the costs and the returns so we can be sure.

4. **Make sure it is the best option**
   We all live in a world of multiple urgent requirements competing for limited funds. (If you don’t have this issue, please write to me and tell me what you’re doing.) What many organizations forget to do is the last step: an improvement may be a good use of funds with a positive ROI, but we should consider whether it

is the best use of those funds right now. This means looking at all your proposed investments as a portfolio and balancing your decisions to come up with the optimum use of your resources. This doesn't need to be complicated either: a healthy debate over the portfolio between the key people in your organization should flush out all the information for the executive to decide.

Think of your best option as multiple filters:

1. Use simple models limited to the most important concepts
2. Only look at what addresses your core business goals
3. Check that the costs are paid back within a reasonable time
4. Make best use of available resources

The work that drops out of the bottom of those sieves will be simple, practical, useful and worthwhile. It is OK for small and medium businesses to take more risks than larger organizations, when we make a considered business decision to do so, in order to focus our resources and efforts. But once a filter tells you that improving your process will advance your goals and is the best use of those resources, do it. In upcoming sections, we'll talk about how.

Practice 2: Delivering to your customers

The most important thing you do is delivering services to your customers. That's the IT service management (SM) perspective. In fact, everything you do should be considered in terms of those services you provide to your customers.

Whether you are in manufacturing, trades, retail, IT, not-for-profit; whether you provide service internally to your organization or externally to paying customers; whether you work anywhere from a small business to a government department; you do service management.

In this century, if you run a business it is most likely a service business whether you know it or not. Customers no longer want to buy something. They don't simply want something done. They want to have a nice easy experience with added value – to be served.

Whether you build roads or map them, operate ports or use them, build houses or sell them, plan weddings or sing at them, care for kids or clothe them, sell PCs or scrap them, you are in a service business, even if you may not be in a "service industry."

We're not talking about over-the-counter "may I help you?" service, the focus of numerous books. Those manuals tell you how to develop the customer service interface, the experience of contact. Instead, service management is about the end-to-end process of providing services.

The customer-centric view

Adopting a service management approach can have a profound effect on the way your business works and your staff think.

It takes us away from that introverted, bottom-up thinking that begins with what we have and what we do and eventually works its way up and out to what we deliver to the customer. Instead, with service management we focus on what “comes out of the pipe” – what we provide to the customer or end user. We take an “outside-in” view. Starting from this external perspective, we then work our way top-down into the service organization to derive what we need and what we have to do in order to provide that service.

Service management isn’t one subset of the business; it is not one activity at the end of the main supply chain. It’s an entirely different way of seeing the whole supply chain, the whole business that produces the services, by seeing it initially from the outside, **from the customer’s point of view**. Therefore our blog posts may stray into general business management topics.

Seeing our business in terms of the services it provides can’t help but make us better at providing them.

To a customer, “better” means more useful and more reliable, i.e., more valuable and better quality. From the service-provider’s point of view, “better” means more effective and more efficient, i.e., better results and cheaper.

Why service improvement?

Improving services gives you an opportunity to increase revenues and profitability, and it brings increased efficiency and effectiveness. That means **increased returns for much less investment** than from improving your products or equipment.

Service improvement keeps your organization competitive. In our service-oriented century, competitors are differentiating themselves on service, and customers are choosing (and staying) based on service.

Service improvement drives service culture. It gets your whole organization working “outside-in”: talking to customers, understanding customers, thinking about yourselves in customer terms, seeing yourselves as customers see you, giving customers service that they want.

Transforming existing service management

You don’t “do” or implement” or “create” service management, even though we use phrases like these all the time. You transform or improve it. Service management is already there in your organization. Perhaps it’s done badly or so little it is undetectable, but it’s there. Your goal should be to build on what is already there, improving and increasing capability.

This brings us to an important point: don’t structure what you improve around service management theory. For example, don’t start a “change” project to implement change control; create a “reliability” project to improve availability of services by screwing them up less often. Include some “change” theory as needed. **Shape the project around the outcome not the theory**. Mix bits of theory where you need them to get to the outcome.

To continue our example, “change” is a broad topic: if you try to “do change,” you will add work that is not directly contributing to the business outcome of reliability. It is better to take bits of change theory and bits from other theoretical areas too and put them together into a solution for what you need right now. If services are unreliable you might also improve tracking and fixing problems and improve availability planning, none of which are Change activities.

Another example: if customers are unhappy with the support they are getting, you might act in several different areas:

- Tighten up response. Look for the worst metrics: maximum wait time to answer phone calls, percentage of requests resolved in first contact, average lifetime of responses…
- Improve reporting of service levels: replace perception with reality (good or bad) then show real improvement over time.
- Reset expectations. Confirm agreed service levels with the customer then communicate to all consumers.

Even though I will describe the key service management processes in future posts, that doesn’t mean you need to adopt all of them right away.

Plan an approach to the transformation. Include steps to address people, practices and things. If the planned cost and effort is not spread equally across each of those three aspects that would be a cause for concern.

Don’t let management design improvements alone. Senior leaders bring context, strategy, and customer needs. Frontline workers bring their knowledge of how to improve operational processes and procedures. And external consultants bring experience, expertise, ideas from outside and theoretical best practices.

Service management practices

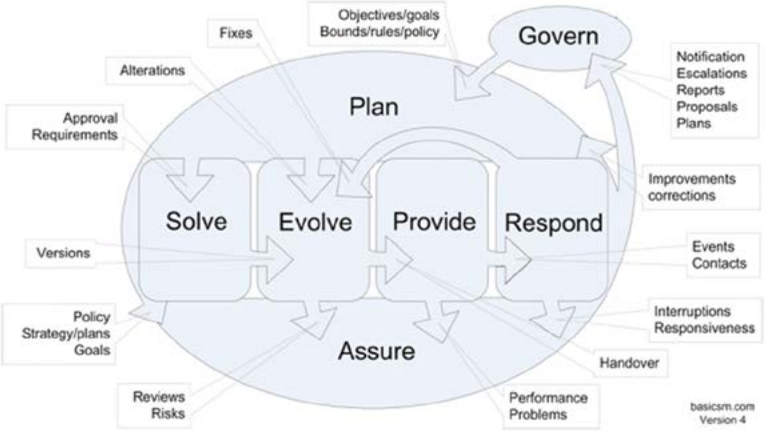
There are several sources of service management theory. They all structure service management – slice it up – in different ways.

We'll divide service management practices into 7 necessary areas of activity:

1. **Plan** how you will run things and what you need to do
2. **Solve** your customers' need: create a service
3. **Evolve** the way things work in your business in a controlled way
4. **Provide** the services to customers
5. **Assure** the services meet goals, requirements and regulations and are safe for you and your customers
6. **Respond** when things happen, especially when your customers ask for advice or assistance
7. **Govern** the system to ensure it meets objectives and stays within the bounds that have been set

The Plan and Assure domains are managing, Govern is … well … governing, and the other four are doing. These are my seven names for these key areas – other theoretical frameworks use different names and slice it differently. I've tried to keep the model generic. It leads into all the other bodies of SM knowledge: it is compatible; it won't lead you in the wrong direction if you want to get into something deeper.

Seven necessary areas of service management



Practice 3: Responding to requests

Look at the chart of the “Seven Necessary Areas of Service Management,” which I just presented in Practice 2: Delivering to your customers. We are now turning to the Respond part: when you are providing services to your customers, it is essential that you respond when they contact you (Contacts) and when something happens in your environment that needs to be dealt with (Events).

In GoToAssist Service Desk, everything we respond to is called an Incident. Some incidents are requests from users, some incidents are internal events. We are looking today at requests.

When you are first starting out, dealing with requests can be simple. In smaller enterprises, you may never need to make it any more complex. Here is a good initial approach to managing requests. Implement these three capabilities one at a time: 1. Record, 2. Respond and 3. Report

1. Record

Provide a single point of contact (one or more people acting as a service desk) with multiple channels to access it. Discourage users from contacting specific people. Steer them towards the service desk, and reward them for using it by giving them good enough service that they will use it again. Recognize support work only if recorded by the service desk: that will motivate staff to steer users there.

Make sure you **keep a record of all requests** as Incidents in Service Desk. Track all your responses and record what you did about them and close them off (tell the user!). You need to manage staff to make sure they do this consistently.

Record all interactions with your users – whether by phone, email, Twitter or accosted in the hallway – as comments against the related incident record in Service Desk. They may call several times about the same thing. It is important to people that you remember the last time they contacted you.

2. Respond

Once you have good records of the requests you are dealing with, start getting smart about how you handle them:

- Make sure someone owns every request (and only one person) and you can tell who that is.
- Match request to other request to see what works and to recognize patterns.
- Build up recorded information on the services and systems. Give responding staff access to information and training, which is kept in the Service Desk knowledge base.
- Use external information: use search engines, get training, get involved in communities and consult experts.
- Provide models or scripts for how to deal with common requests.
- Use Service Desk to pass requests to someone else. That someone might be a specialist in dealing with it or a boss because it is happening too slowly. The specialist might be in another organization, which usually means email as the way of passing it to them, which has a risk of requests getting lost, so have crosschecks and follow-up. Better still, persuade them to use your Service Desk system to manage and record what they are doing for you so everybody can see it in one place.
- Regularly monitor how long requests are taking and chase up the slow ones.

Try applying a little psychology to your interactions with users. These principles have proven to be effective. (They come from “Using Behavioral Science to Improve the Customer Experience,” J. DeVine and K. Gilson, *McKinsey Quarterly 2010*.) Train your staff in them:

- Get bad experiences over with early: talk about the difficult parts first.
- Break up pleasure and combine pain: bring all the unpleasant bits together. Sprinkle the good news throughout the discussion.
- Finish strongly: have a positive (scripted?) finish, emphasizing the benefits to the consumer.
- Give consumers choice: allow them to be in control as much as possible. Steer them but give them their rights.
- Let consumers stick to their habits. Don’t force change unless absolutely necessary. If the old way is good enough, leave it alone. When they must change, ease them across gradually.

3. Respond

All those request records stop you from dropping the ball. They help keep you organized and prioritized.

But the other big payoff is **analyzing the data to look for trends**. What areas generate the most questions (training may be required), which users complain the most, which services have been flakiest, which staff are extra efficient (learn from them) or not so efficient (help them improve): there is great value there to help you enhance your service.

There you go: **three simple phases to setting up your request process** on Service Desk:

1. First get staff to record every request
2. Then get organized at managing the response to the requests
3. Finally start using the data to improve

Practice 4: Response to incidents

In my previous post in this Sensible Service Management Series, I looked at the core of servicing customers: managing Requests.

In GoToAssist Service Desk software, everything we respond to is called an Incident. So, let's talk about Incidents in the strict sense of the word: dealing with things going wrong.

(One caveat first. ITIL confuses things by talking a lot about finding and fixing the **underlying Problem**, and recovering the broken service, as part of the Incident process. I'm going to analyze those separately as part of the Problem process in Practice 6. Here's how I want to distinguish Problems and Incidents:

- **Incident process** is about getting the user(s) working again as quickly as possible, however we manage that.
- **Problem process** is about fixing the underlying causes.)

Now with respect to Incidents, everything I said last time about Requests still applies. But I want to add a few more considerations when there's an Incident needing to be fixed. As you'll recall, **there are three main capabilities:**

1. Record

Provide a single point of contact with multiple channels to access it. Make sure you keep a record of all requests as Incidents in Service Desk. Record all interactions with your users. Track all your responses and record what you did about them.

2. Respond

Make sure you own every request. Use Service Desk to track and organize the workflow, passing requests to the right technician. Regularly monitor how long requests are taking and chase up the slow ones. And employ Service Desk's capabilities to make support more efficient and reduce demand on your team. Provide scripts for how to deal with common requests. Build up information in the Service Desk knowledge base for your technicians and for end-user self-service.

3. Report

Look for trends to help you improve your service.

OK, so now let's elaborate on those three main capabilities. Sometimes a user requests help with a service not working as expected — something needs to be fixed. That is an "Incident" in the strictest use of the word.

That "user" reporting an Incident may be an internal person picking up an error before it affects any of the "real" users consuming the service. It can even be a software program detecting the error and automatically alerting us.

Responding

If something needs fixing, then 2. "Respond" is obviously a crucial part of our Request process and how we resolve the incident. So, I want to expand my analysis of this core aspect.

2.1. Categorize

It helps to categorize all incidents, whether general requests or issues to be fixed, but it is particularly important to get a general idea of what type of incident it is. We need to determine how serious it is and how wide and severe is its impact, so that we pass it to the right person the first time.

2.2. Diagnose

This is where keeping records of past incidents and responses, along with building up the knowledge base, really pay off. Search Service Desk's Incident records, Problem records and the knowledge base to see if we know what the cause is and how to fix it. If you get a match, fix it or pass it to somebody who can. This is called level 1 support.

If you don't get a match or can't fix it, pass it to level 2 support: those who have the technical skills to do specialist diagnosis and resolution. If they can't fix things, they refer the incident to Level 3 support: the folk who built or supplied the stuff that is not working, often a supplier external to your organization.

2.3. Escalate

All this passing around among support groups is called "functional escalation," but when we talk of "escalating" we usually think of "hierarchal escalation," i.e. telling somebody more senior. We hierarchically escalate because:

- The incident impact is serious enough that they should know about it
- A fix can't be found

- Someone is not responding fast or well enough considering the severity of the incident

That senior person might determine that this is a “**Major Incident.**” This means we drop the normal process described here and switch to a crisis-response process that will be described in a future blog post.

2.4. Resolve

Somebody is not getting the service they expect. The incident process must focus on restoring that service. That might not be the same thing as fixing the underlying Problem. If we have to fix the Problem in order to get the user back on track, we will, but sometimes there is a “workaround”: a way to get them back up and running without fixing anything. For example, with some software, simply logging off and on again may get them around an issue and working again. Or rebooting a server may make the problem go away. (There’s an old IT joke: “A problem gone is a problem solved.”)

You can find workarounds in Service Desk’s Problem records and/or you can also record workarounds in the **knowledge base**.

Eventually a Problem may cause so many Incidents that we have to hold the user up without a workaround while we properly diagnose it and nail it once and for all. That is a management call whether the inconvenience is outweighed by the ongoing cost of recurring incidents. But in general the Incident process takes whatever workarounds or temporary fixes it can to get service restored to the user as quickly as possible.

2.5. Close

This applies to all Incidents and Requests. Before you close the ticket make sure:

- The incident impact is serious enough that they should know about it
- You tell the user it is done
- The user thinks so too and they are happy with the outcome
- The Incident is properly categorized so our reporting data is useful
- The Incident has a record of everything that happened and what workaround or fix you used. In the future, you or one of your colleagues may be grateful you wrote it down.

There is a huge body of knowledge out there about Incidents and Requests, which you can investigate further as you need to. ITIL has a lot (in the version 3 book Service Operation and the Operational Support and Analysis intermediate course). The Helpdesk Institute (HDI) produces a lot of useful material too. COBIT 5 is my choice for formal definition of what should be happening and what should be produced and by whom.

For now, start with:

1. Record
2. Respond

2.1 Categorize

2.2 Diagnose

2.3 Escalate

2.4 Resolve

2.5 Close

3. Report

Practice 5: Dealing with major incidents

Up to this point in the Sensible Service Management Series, we have looked at service management in general and then at how to respond to incidents and requests. That responding activity was all about the day-to-day activities when things are close to normal. But sometimes things get a long way from normal. I'm talking about when the proverbial hits the fan. Something is so catastrophically broken that we need to drop normal processing and switch to a crisis mode. **This is known – a bit prosaically – as a Major Incident**.

In some schools of thought – me for instance – you can also have a Major Problem. They are different paths to the same point: we respond in the same way.

The first time we have a Major Incident, the response is usually to panic. Or at least, a lot of turmoil ensues, as people rush around working out what to do and who should do it. We get things fixed quicker and we keep our customers happier if we have planned how to deal with Major Incidents. Yes, **we need to think about Major Incident Management (MIM) before catastrophe strikes**.

Setting the scope for MIM – A Major Incident is more than a high severity incident, which we give our highest priority and put our best people on it, yet we still use the normal procedures to deal with it. And it is something less than a total disaster, where our systems are wiped out and we have to go to our disaster recovery (DR) plan. A Major Incident sits somewhere in between. Sometimes a high severity incident turns in to a Major Incident, when we realize the impact is worse than we thought or when it takes too long to fix and there is no sign of a resolution. And sometimes a Major Incident turns out to be so bad that we give up trying to deal with it and declare a disaster. So there's a ladder of incidents with increasing severity:

Incident	Normal procedure
High Severity Incident	Normal procedure with top priority
Major Incident	MIM procedure
Disaster	DR plan

(Notice that I refer to a DR plan, not a DR procedure. In a disaster, we plan the things that need to be done. For a rebuild of our systems, we often define a series of steps that have to happen in a certain order, but there isn't so much a procedure for dealing with a disaster (do this, then this, then this) as lists and policies. We have lists of the systems we need to restore first, lists of contacts and so on. We have policy that sets the rules and bounds and trigger points. **But there is little point in trying to create a defined procedure because the circumstances of every disaster differ so much**. We are going to have to make it up to suit the situation. The DR Plan helps us be as prepared as possible.)

MIM sits somewhere in between a structured incident procedure and the guidance of a DR plan. In a major incident, we can say what the initial procedure will be, but beyond that, it will again vary depending on the circumstances. So, we also need to have **a MIM plan to provide as much guidance and information as possible**. The MIM plan should define the following:

Policy

Some organizations put a lot of effort into defining what constitutes a major incident. It is futile to define the exact measures of one (more than 90% of this; three or more of those). Trust me, the real world will invent a new crisis that is outside your definition yet clearly a Major Incident. It is worthwhile defining guidelines or principles for recognizing a Major Incident, but a Major Incident is like art: hard to define but you know it when you see it. And like art, you need a human to recognize it. So t**he key thing to define is not what a Major Incident is but who gets to declare one**. This should be equally broad: any manager above a certain level, say. It must be easy to find a qualifying person in the heat of a crisis.

Other policy sets the principles we work by in addressing an incident, the goals, the rules and bounds, the responsibilities.

Roles

There is a **Major Incident Manager role**. You need several designated people willing to take on this role in a crisis. They may not be the same person as your day-to-day incident process manager. Choose people who are natural leaders, strong communicators and good under pressure.

Other roles include a Communications Manager, to deal with internal and especially external communication, and a Resolution Manager who directs, understands and coordinates the technical people fixing the problem. You need several designated people willing to take on those roles too.

Procedure

The first few steps in dealing with a Major Incident will be the same in almost all cases. By having these defined, we bootstrap ourselves up to a state where we can start writing the rest of the procedure on the fly. These steps are listed in a checklist I posted at www.basicsm.com/declare-major-incident-or-problem.

Once we have a Major Incident Manager in control of a center of operations, we are in a situation to plan the next steps according to the situation, to hopefully resolve the incident. Those steps will include some standard ones which should also be documented in the MIM Plan:

- The Major Incident Manager reconfirms that there is indeed a Major Incident
- The Communications Manager communicates the schedule and methodology for future updates to all parties
- The Communications Manager notifies all business owners of the service(s) and other stakeholders listed in the Communications Plan
- A Resolver Team or teams are assembled by the Resolution Manager. The Resolver Teams agree to their communication schedules and protocols before getting to work.

Communication Plan

The Communication Plan should describe who needs to know what, how, and how often – from customers to non-involved internal staff.

It should describe what happens at certain points:

- Regular updates from the Resolution Manager to the Major Incident Manager. Nobody else communicates with Resolver Teams except the Resolution Manager. Leave them alone to get the job done.
- Regular updates from the Communications Manager to stakeholders
- Meetings of the Major Incident Manager and Communications Manager with senior internal and customer management and other stakeholders

Center of operations

The MIM Plan should describe how to set up and run a “war room” for the duration of the incident.

This includes looking after people: feeding them and making sure there are rosters of staff so that they get some rest, especially the key people. Therefore you need more than one person for each of the key roles. After 12-18 hours under pressure, people are dangerous decision-makers.

If this MIM Plan seems like a lot, it won’t be when the day comes, as it most certainly will. You will wish you had planned more.

One last point: rehearse this just as you rehearse fire drills. Rehearse the initiation until the center of operations is up and running. Rehearse the meetings with stakeholders. And rehearse the root cause analysis and problem resolution. We’ll talk about those last two next time when we look at Problem Management.

Some of the content in this article comes from the MIM checklists at www.basicsm.com/checklists. You can use those checklists to help keep control in a crisis. (Braun Tacon of majorincidenthandling.com contributed to the MIM checklists.)

Practice 6: Problem management

So far, the Sensible Service Management Series has covered incidents and requests. This is the “front-office” activity involved in serving the users: meeting their needs and keeping their services running. There is a “back-office” activity closely related to incidents and requests: **Problem Management. A problem is an underlying cause of incidents.** Usually it means something is broken. If an alligator bites someone, you fix the incident with bandages and maybe surgery. To fix the problem, you shoot the alligator before it bites again.

Strictly speaking something can be wrong/broken and not be a problem because it is not causing incidents (yet). I like to call those Faults. You can keep life simple (and your GoToAssist Service Desk configuration simple) if you treat anything wrong/broken as a problem.

It is not uncommon to find an organization that doesn’t use problem records, only incidents. This is a big mistake. An incident record says that a user is unhappy. If we get the user working again (say using a workaround – see Practice 4: Response to Incidents) then the incident is over, though the problem may be still there. When an incident ends up getting used to track the problem, this screws up our reporting, making it look like we have long-running incidents and that we are not looking after the users.

Incident Management and Problem Management are very different activities and need to be kept separate. Not only does it make the incident reporting more accurate; **keeping problems separate has other benefits**:

- We can see our “portfolio” of problems: the overall situation gains visibility, so we can prioritize what we need to fix and work out how much resource we need.
- Incident and problem practices often have conflicting objectives. For example rebooting a server will quickly fix a lot of incidents but it potentially destroys diagnostic information for resolving the underlying problem. These conflicts should not be laid on one person to reconcile. Different groups should make their case to higher management to resolve it when they conflict.

So use problem records. **We open problem records in several different situations**:

- There is an incident and we can see there is an underlying technical cause (even if we don’t yet know exactly what it isn’t, say, a user error or an administration mistake)
- We detect a pattern in incidents and start to suspect there is an underlying cause for them
- We see something is wrong/broken

If you want, you can be quite general about what you define as a problem. For example, lots of user errors might show you there is a problem with the training.

Track all your problems (prioritize, work on them, follow up the slow ones), and record what you did about them, **and close them off** as you fix them or decide to live with them (if they are too hard or expensive to fix).

It is not the boss’s job to solve problems. Problems don’t get escalated. The old manager’s mantra is “Bring me options not problems.” Those doing operations know best how to fix problems.

In order to fix a problem (or an incident) you quite often have to do **root cause analysis**. There are formal techniques you can use to do this. Some argue that there is no single root cause of problems. It generally takes several causes together to create a problem – they have to "line up" in some way. The first and most obvious cause you find is seldom the end of the story: keep asking "why" until the answers are not useful. Finding root cause is not necessarily about assigning blame – it is about removing cause. Complex systems are in fact permanently broken, so when they actually fail, it may be nobody's fault. On the other hand, there could be negligence.

Once you are tracking and dealing with problems, the next level of maturity is to "kill the alligators before they bite you": **proactively seek out problems** and fix them. When you are really good, you will forestall them and prevent them ever existing. Find a keen, clever, energetic employee and assign them half a day per week to be an Alligator Killer: measure their success on how many problems they find and eliminate.

Your register of problems in GoToAssist Service Desk is closely linked to your register of risks, and you may want to link them. An unfixed problem poses the risk of future service interruptions.

The better you get at dealing with problems, the fewer incidents you will have. The other area that you can improve in order to reduce incidents is Change Management. We'll talk about that next.

Practice 7: Change management

In our Sensible Service Management Series, we've considered the "front-office" activities that serve the users in chapters on "Responding to Incidents" and "Responding to Requests", and we've explored the "back-office" activity that removes the causes of incidents: "Problem Management." If an alligator bites someone, you fix the incident with bandages and maybe surgery. To fix the problem, you shoot the alligator.

But how about a fence to keep alligators out in the first place? This might be considered **Change Management**, which is another "back-office" activity and the topic of this chapter.

All changes should be managed together as a portfolio, as part of planning. They should be balanced against each other and against available capacity and capability. At some point, all changes — large or small, organizational or operational — move from designing and building into a live or "production" state. **That movement of changes needs to be controlled to minimize risk and protect operations**. That is what Change Management (in our narrow sense) deals with: managing the change to the operational environment.

Change need to be managed. That sounds obvious but many organizations balk at spending the money for change and project managers. Managing a change takes time and skills, and the investment of that extra 10% pays off in changes being successful and on time and on budget.

Project management

For any change above quite a small threshold, **make it into a formal project:** plan and control it. That threshold should be based on risk and potential impact, not just size.

On any but the smallest projects, a good **professional project manager** repays their overhead by ensuring the costs, risks and over-runs are controlled in the project.

Every project should have **a steering committee**, with a single decision-making executive in charge: they're the "owner" of the project. The manager running the project should be separate from the steering committee and report to them.

Manage a large or risky project **in stages**. At the end of every stage, you should reassess the justification and the plan: Do you proceed? What should you adjust? At the start of each stage, the owner commits the next portion of resources (funds, people, etc..), and the project manager plans how to use them to deliver the outcomes of that stage.

Every project should **include assurance**: either the steering committee or auditors they appoint should both challenge and advise those working on the project to assure that policies, rules, plans, specifications and standards are being followed.

Every project should **set up what comes after** it: ongoing ownership, operation and maintenance of what was built; ongoing improvement; ensuring the benefits are realized; and measuring the actual results over time.

Production

Put a “fence” around everything that provides services to consumers; i.e., define a boundary and call it your “Production” area. Restrict who can change Production. This includes facilities, equipment, stock, software, documentation and spares.

Get some control over all the changes to Production. Define what is considered administration and therefore not covered by change-control. Administration activities should still be access-controlled and audit-trailed.

Change control

To start with, make sure all changes get recorded somewhere central.

Later, make sure they are recorded beforehand and that the schedule is planned. Is this a good time? What else is going on? Can we batch these changes up and do them together? Can we have a regular time we do this stuff? Publish the upcoming schedule regularly: usually weekly, sometimes real-time.

Once you plan your changes in advance, set up a way to approve all changes. There is a set of approvals required, depending on the size and complexity of the organization:

- Business owner of the change
- Operations
- Change coordinator

In addition to the approvals, many organizations will want to set up a group of stakeholders to review changes, an “advisory board,” to make sure all implications have been considered. Make a list of the people they should consult depending on the impact of the change.

Don’t approve a change unless you understand the risks and the operators know how they are going to back the change out or restore things if it goes wrong. Do some policing to make sure folk aren’t sneaking changes in.

Once all changes are approved and nobody is subverting the controls, define (carefully) what types of operational changes are “standard”: they don’t need approval because they are low risk (not all easy or small changes are low risk and not all low risk changes are small or easy). They still need to be written down and often need to be scheduled too.

Require that a change is ready before it gets the OK to go into production.

Has it been tested?

If it is a new system/product, have we worked out how to support it? Trained the staff, procedures to operate it, procedures to fix it, vendor support contract?

Have we got enough information about it? Manuals, documentation, vendor details?

Environment

Which brings us to the next point: know what you are changing. A good start is to **know about all the things in your environment**. Some of these are assets, which crudely are the things that come through procurement – so that is a good source to start with in listing all the things in the production environment.

Some things are not assets in the sense of what you buy (and depreciate), such as documents, people and places. Other things are conceptual or virtual, such as systems or the services themselves.

One way to record things is to capture them as they come into production, like the procurement example, or like all the information about the things in a project that gets rolled out. In theory, if you do Change properly you will know about them all. But in reality, this is usually a “leaky” approach: things get into production undetected, e.g. IT devices like laptops and wireless modems. So at intervals, you also need to discover or audit or take stock of what is in production to check against what you think is there.

Release and deployment

Often your services will involve a complex mix of people, processes, plant, logistics and computing. It is not enough just to schedule changes: you may need to manage their impact on the service, by

- Considering how the changes interact
- Bundling changes together into a package or “release” or “version” of the service
- Ensuring the target environment is ready (people trained, infrastructure compatible, operational procedures adapted…)

- Managing the steps in deploying the change(s)
- This higher level of management for rolling out complex changes is called Release (and Deployment) Management. We'll talk about it next.

Review

Follow up on changes.

- Did they work?
- Did we get the results we expected?
- Did we manage them in the right way with the right level of formality and controls?
- What can be learned for next time?
- Did they get all approvals they needed?

A successful change is one that achieved what it said it would, not a change that didn't break anything.

Dissect changes that fail. Try to identify the causes of failure and prevent a recurrence. Most accidents have multiple causes acting in conjunction, so look for more than one.

Formal projects should get additional follow-up some time after completion to ensure they delivered the return they said they would in the business case for doing the project. If projects didn't deliver as expected, then look at how business cases are written, how the decisions are made, how projects are managed and how the value is measured.

Practice 8: Release management

Previously in this Sensible Service Management Series, we looked at Change Management. Part of any change is rolling it out to your production environment. Sometimes a change means a single update to one piece of equipment or software on one computer. We manage the implementation into production simply as a step in the change process.

Other times, the change needs to be deployed to many sites or to everyone's desktop computer. Now that deployment is a whole new ballgame. **When the complexity and/or risk of going live with a change reaches a certain level, we introduce a new discipline to manage the deployment: Release and Deployment Management**.

In yet another scenario, we might have many changes to deploy, none of them necessarily complex on their own but all targeting the same area, for example the same software. So to reduce the effort and risk, we **bundle all the changes together into what is called a release**. The release will deploy what is often referred to as a new version of the target component.

Sometimes we choose to bundle changes into a release because the changes affect different but related target components. For example, we upgrade some software on desktops and that also requires replacing some of the computers with newer machines, upgrading the operating system on others and making changes to the network and a database on a server. All the changes have to happen together because they all depend on each other. They are rolled out as a single release.

And of course sometimes we do all of these at once: deploy a bundled release of changes to a complex distributed target environment. This often happens when we make a major upgrade to a service or introduce an entirely new service.

Have a formal project plan for any release. About a third of the effort should go on people-oriented activities, one third on practices, and one third on technology/facilities/things. It is OK to diverge from that, but it is also OK to ask why.

Have **a formal handover** of a release to production operations. Those taking responsibility for it have a right to assess it before accepting it. We'll talk more about new services and about production readiness another time.

For major releases, those creating it should help run and **support it for a warranty period**. Before the project is disbanded and the builders all run off, ensure enough information is documented and shared to be able to support, fix and change the new system.

For any release – heck, for any change at all – make sure you **have the documentation** of what changed. For some changes, such as software, make sure you have a definitive copy stored away somewhere as the master. For other changes such as new equipment, make sure you have the **necessary spares** locked away for when you need them.

Equally important, **ensure enough knowledge of the system has passed** to the team of those who will have to support, fix and change it. The best way to do this is to have operational people work on the design and build of the service (or builders who go on to work in operations). Training at handover is a poor substitute.

Right back at the start of this eBook, we said small organizations focus on what matters: successful daily operations, growing the business, controlling costs and keeping customers happy; and we said you will deliver on these goals when you improve (1) how you respond to user requests, (2) how you fix things and (3) how you control changes to make sure you don't break things in the first place. Now we have explained how to do that: (1) Request Management (2) Incident and Problem Management and (3) Change and Release Management.

Look at the picture of service management that we gave you in our second chapter Service Management Success: Delivering to Your Customers. We have talked about the two main control areas: Evolve (Change, Release and Deploy) and Respond (Incident, Problem and Request). These are the areas where GoToAssist Service Desk can help you most.

Practice 9: Deciding to improve service performance

There's a common assumption that everyone wants to improve service. Infinitely. But that is not always the case. So, don't automatically leap to the assumption that you need to improve. **That is a business decision like any other**.

The first consideration is **whether you really need to improve**. Some questions:

- Are you in an industry where there's no expectation of high levels of service, where service is not a competitive factor?
- Do you have captive customers, where you don't need to keep them happy? (Warning! Validate these first two assumptions regularly: things change, competitors enter the market, new options open up.)
- Can you choose to compete on something other than service, e.g., price?
- Are you battling to survive and can't afford higher levels of service right now?
- Are you already as good as you need to be? (More danger! Keep measuring to ensure that stays true.)

In other words, improving service – and which services to improve – is **a strategic decision**. You decide to make improvements **because it fits your business strategy**.

Once you have decided you need improvement, you must consider **what constitutes "better."** (We'll talk about measuring service performance next chapter.) For now, when assessing how good your service is, you should measure it from the "outside inwards." **Measure how your service looks from the outside**.

There are two main perspectives to this: (1) how you look to your customers, measured by things your customers care about such as quality and user satisfaction; and (2) how you look relative to your competitors – if you are ahead of the pack then you may be able to put resources to more effective use than in getting even better at service.

If you define what better looks like, then you **define the business outcomes you want** from improvements. From those outcomes, you then decide a shortlist of improvements you could make in order to deliver those outcomes. These can range from introducing a whole new service (or killing one off) to a major upgrade to minor tweaks. You can make improvements to the design of your services, the infrastructure that delivers them, the way you execute them or the way you run and support them.

We'll talk later about how to decide what you are going to work on. Usually if you haven't improved for a while or there is lots of room for improvement, then the first things to work on will be pretty obvious. Ask your users, customers and staff – they'll be sure to tell you.

The next consideration is to determine or **estimate the benefits of those improvements**. What is it worth to you if your service is better? The benefits might be higher levels of customer retention, more new customers, more repeat sales, more follow-on up-selling or reduced cost of support. Those are value benefits.

The other side of the benefits coin is **reduction of risk**. It can be a lot harder to put a number on what a reduced risk is worth to you. Sometimes you can find a dollar value for a reduced or removed risk, other times it is an unquantified benefit, and sometimes it is non-negotiable, e.g., your auditors say you absolutely have to remove the risk.

Another consideration is **what it will cost to improve**. This may have to be a “wet finger” estimate for each proposed improvement. There are many things we need to improve in any organization; if you try to plan and estimate each one in detail, you will not be able to afford to get them done. The hard reality of most enterprises is that a lot of improvement has to be done informally, loosely managed as part of daily business, or it will never get done at all. If the improvement is large scale, complex or high risk, then you should of course plan and manage it more formally using project management methods, and you will get a more accurate estimate of the costs involved.

Finally, you need to look **at the business case for making improvements**: weighing the benefits against the costs. There are improvements that your customers might be crying out for, or that look like a great idea, or that your staff are saying “How can we not…?” but the improvements just don’t pay – they are a bad business decision. And there comes a point of diminishing returns where you have made all the high-value improvements.

Don’t look at the technology costs alone. These are only one part of any improvement. Far more important (as we will see in a future post) is that any improvement is always trying to **change the way people behave**: staff, customers, and/or users. Behavioral change involves changing people and processes. This makes changing the technology look simple and cheap, but if you scrimp on behavioral change then any technology investment will end up wasted because you won’t achieve your ultimate aim.

Costs to consider:

- Communicating and consulting on new goals and systems
- Designing, testing and documenting new procedures
- Training in new systems (procedures and tools)
- On-going support for new systems
- On-going maintenance, repair and improvement of new systems

A good rule of thumb is to work out the technology spend then estimate as much again for each of process and people costs, i.e. triple it. **Anything less than doubling the technology costs is deluding yourself**.

Weighed against these costs are the benefits, tangible and intangible: money saved, potential new revenues opened up, risks reduced, compliance assured, strategy delivered.

Then recall what we said right back in the first chapter of this series:

> What many organizations forget to do is the last step: an improvement may be a good use of funds with a positive ROI, but we should **consider whether it is the *best use*** of those funds right now. This means looking at all your proposed investments as a portfolio and balancing your decisions to come up with the optimum use of your resources.

If you are a small business, then you likely won't go through all this analysis. You'll make a gut call that you need better service. The best way to inform that type of decision is to **talk to your customers** as widely and as often as possible. Don't stick to the customers who are happy to talk to you, that you are comfortable with. Reach out to others, especially ones that are hard to get to: this may mean they have a problem with your service that you need to know about.

To summarize:

- Decide if your organization needs to make improvements to your services. This is a strategic decision taken at an appropriately senior level.
- Decide what better looks like in business terms, what the desired outcomes are, while looking at your service from the outside.
- Choose a shortlist of improvements.
- Work out the benefits (value, savings, reduced risk) of each.
- Weigh that up against the costs of each to decide which ones make business sense.
- Improve.
- Repeat.

Practice 10: Measuring service improvements

In the last chapter, I talked about whether you actually want to improve service at all – it's not necessarily a given that you do. And we reviewed the essential question: how service improvements fit into your business strategy.

Now, that we've made the business case that service improvement is needed and worthwhile, here's the next question to consider: **what constitutes "better"**?

Better

There are three kinds of "better" service:

- A specific measurable goal: e.g., cut costs
- A broad goal: e.g., make customers happier
- The goal to just keep improving service

Depending on where you fit on that spectrum of goals, your answer to the next question will change: how will you know that you have improved?

There are two possible answers:

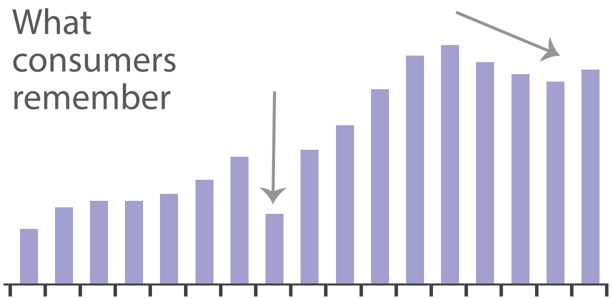
1. By measuring the difference in something
2. Subjectively

This may seem blindingly obvious and subjective assessment may seem like a bad idea, but think for a moment. Because to measure the difference in something you need:

- To decide what is a good measure or measures for what you want to improve
- A mechanism for measuring, recording and reporting
- To do the work to measure the current state
- To do the work to measure again afterwards and work out the difference

This could be an expensive exercise. What is the purpose of improving? Often (but not always) it is to make customers, users and/or managers feel better about the services. (Or it could be to cut costs, speed delivery or some more objective goal.) If the goal is to make people happier, then why not just ask them if they are happier afterwards? If they aren't feeling happier, then all your improvement metrics could be a waste of time.

On the other hand, good metrics may help change their minds and convince them to feel better. After all, users most strongly remember the recent past or painful past. Showing objective progress can help put a recent minor slip in standards into context.



So it is management's call: spend on measurement or to subjectively assess. Both approaches have their merits.

Assuming you decide to measure your improvement, the next question is: **what metrics will you measure?**

External

The overall measure of service should usually be **from a customer perspective**: in terms of the business results delivered, not in terms of some internal performance. You need to measure enough to know if you are delivering service well enough – usefulness and reliability – and to know if efforts at improvement had any effect.

The **classic service metric is customer satisfaction**, measured by surveys. This can be challenging to get an accurate measure – it is subjective. It's a useful measure but not usually the best primary measure. Note that customer satisfaction is not always the same thing as user satisfaction; in some cases those paying for the service don't actually care how satisfied the users are with it.

A popular tool for measuring customer satisfaction is NetPromoter. This is a good approach, though you might like to read my sceptical blog on the subject before deciding.

The most obvious objective measure of service is its **availability**: was it available when the customer expected it to be? This seems simple enough until we define "available." If the service is too slow then it is not really available even if the "door is open."

This leads us to **service quality**, the standard of service, the quality of the user experience. The metrics for this will differ depending on the actual service. Some minimum standard must be set, below which the service is deemed unavailable.

Once you get your operational metrics right, and understandable by customers, in some situations you can move up to reporting the customer's processes (and your contribution), and even move up higher again to report the value you provided to the customer. Put another way, report not on what you do but rather how you help them do what they do.

Internal

Of course, your objective in improving service may not be customer oriented. You may seek to optimize service for internal reasons, e.g. lower cost, risk, cycle times, effort, higher quality, agility, accountability, profit. Sometimes the intent is to cut costs by having user satisfaction as low as possible without actually alienating anyone (e.g., after-sales support from some vendors).

Focusing too much on any one metric – whether you are reporting to customers, staff, managers or governors – will lead to distortions of behavior: people get driven to improve the metric even at a cost to something else. Metrics all cause this effect: the perfect metric has never been invented.

One way to reduce this problem is to employ a balanced scorecard (or a performance pyramid, results and determinant matrix or performance prism – there are many alternatives). The classic balanced scorecard has scores for four groups of metrics, typically with half-a-dozen well-chosen metrics in each group, based on the main objectives and strategies. The four groups/dimensions are:

customer	financial
internal business processes	learning and growth / innovation

There are a number of variations, including:

- Dimensions specific to the business or department
- Nested scorecards at strategic and operational levels
- An overall (often weighted) score for each group and trying to improve that score

The original concept of balanced scorecard from Kaplan and Norton had the dimensions above, but other combinations are used. One service-oriented variant you might like to try:

customer value	efficiency
effectiveness	improvement

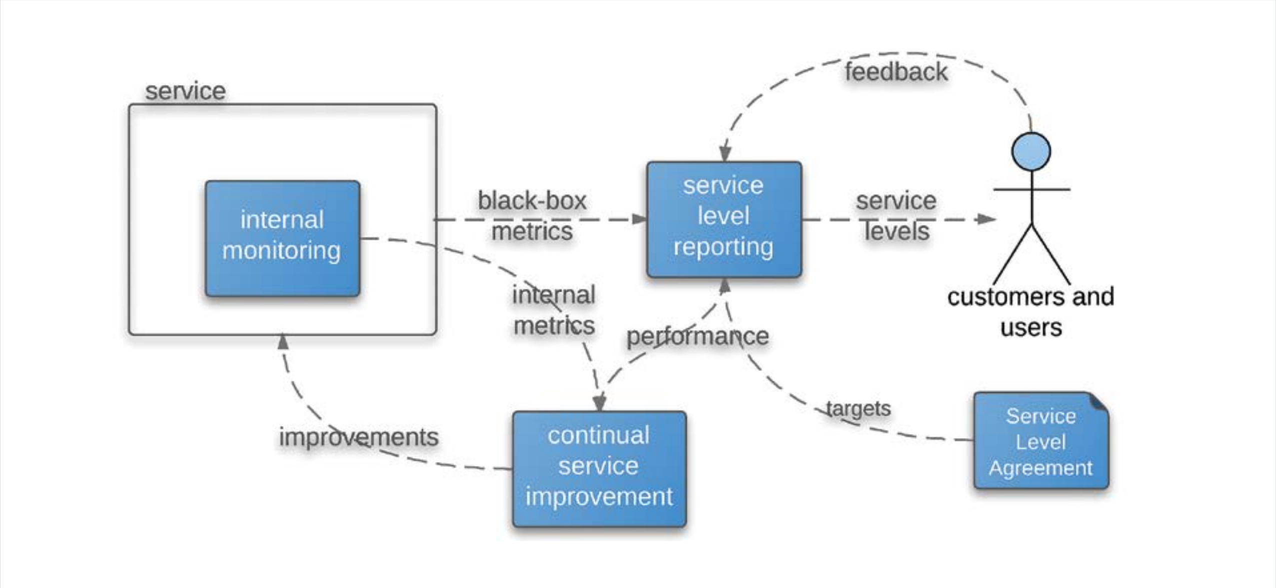
The main point is to look at a **balance of many metrics across different dimensions** instead of making decisions based on one or two numbers.

Another good practice is to always include a commentary with each KPI, an “intelligence report.” Numbers on their own do not tell the whole story, and behind every number is a story: why it is what it is, why it has been changing, what it is not showing…

External and internal metrics working together

So service levels should ideally be measured with customer-centric, outside-in metrics. We can use service level metrics like mean time to resolve incidents or cost of servers to help us optimize the internal “machinery.” They are good for helping improve internally but they don’t measure the service: they don’t measure how happy customers are with the result, what the costs were or how much value it delivered.

Outside-in, black box measurements of service delivery like satisfaction, cost and value will work well as service level targets in a Service Level Agreement (SLA). We combine feedback from the users and how we are tracking against the SLA targets as performance information. We combine that performance information with the internal operational metrics to determine what needs improving.



Measurement is good: it gives us objectivity and makes progress visible. It is also expensive to build and operate, so think carefully about what you really need and why. All measurement should have a purpose.

The primary purpose of measurement is to improve. Next chapter we'll talk about how to improve service. (Portions of this chapter are derived from my books Basic Service Management and Standard+Case.)

Practice 11: Improving service performance

Right back at the start of this series on Sensible Service Management, I noted: SM process improvement can deliver on four top goals of an enterprise:

- Successful daily operations
- Growing the business
- Controlling costs
- And keeping customers happy

You'll deliver on these goals when you work to improve three areas: (1) how you respond to user requests, (2) how you fix things and (3) how you control changes to make sure you don't break things in the first place.

We've already discussed the practices in those three areas, and we've looked at the principles of measurement. Now at last we look at **how to go about improvement**.

Focus on outcomes

Think about the practices and functions we have discussed already that are required for services to be delivered well: service desk, incident, major incident, request, change, release and problem. If any one of these is not being performed, or not to an adequate level of capability, then there is a risk to delivery of services.

Which of these practices and functions require some improvement in your organization? It might be easier to list those that do not require any improvement. There is a lot to consider when improving services.

How then to cover everything that needs to be examined? The key word here is “needs.” We should understand **what are our business goals for service, derive from those goals what are the required outcomes from service delivery, then focus on improvements that deliver those required outcomes …** and nothing else.

One way to improve focus is to work on smaller units than a whole practice. A major shortcoming of many IT service management projects is that they take the ITIL “processes” as the building blocks of the program. “We will do Incident first.” “We can’t do Change until we have done Configuration.” Even some of the official ITIL books promote this thinking.

Put another way, you don’t eat an elephant one leg at a time: you eat it one steak at a time… and one mouthful at a time within the meal. **Decomposes the service management practices into smaller, more achievable units** of work, to assemble Lego-style into a solution to the current need. The approach goes like this:

The value we are asked (or propose) to produce is A.

Therefore the organizational goals or objectives are B.

Therefore the improvement outcomes required are C.

Therefore the program outputs to be produced are D.

These outputs come from practices E, F and G.

Specifically they come from the following bits of those practices H thru Z.

By breaking it down into only the bits that compose a solution to deliver to a business value, rather than working on a whole practice for its own sake, we are **starting on the outside and drilling in, focusing on demonstrable value**.

For example:

Incident Management can be broken down into separate units of work:

- Incident tracking
- Incident impact assessment
- Incident escalation
- Incident resolution
- Incident reviews and reporting
- Incident matching

Some of these have dependencies on others. In other cases, they can be improved independently.

These tasks don’t attempt to be a complete description of Incident Management. There will be other pieces of work we could do later.

This helps us to shortlist improvements and to narrow the scope of work.

Accept risk

Frequently the business goals are general: “improve quality,” “cut costs,” “retain customers”… This throws a wide net, so it is often the case that our “shortlist” is still overwhelming. How to address everything? Don’t!

Every organization carries risk – we live with it, we manage it. Usually, the ideal state is unachievable. Instead, a realistic compromise must be found. This hinges around an acceptable level of known risk from unaddressed requirements.

We must consider available time and resources and then develop a **Continual Service Improvement (CSI) Plan to work on only those tasks with which we can deal given the constraints**. The hard calls have to be made: what is in and what is out. This should be a collaborative exercise; everybody needs to understand the hard decisions that were made.

It helps to sort the tasks by what they deliver: business value, risk reduction and/or compliance.

It is better to manage a known risk caused by an improvement being left to later, than to have an unknown and unmanaged risk resulting from it being neglected. Leaving work to later does not mean these improvements are any less important. All selected improvements are important. A reality check says that they can’t all be done in the immediate future, so we have to defer something. We all defer work anyway: this approach manages it and makes it a known risk instead of turning a blind eye.

Managing the work

When we look at the list of improvements that absolutely must be done and cannot be left for later, how should we approach them?

We could **form a project or projects to make the improvements**. A project approach is structured and controlled and carefully estimates the resource commitment before starting. A project approach also considers the best solution and ensures it is a complete one. Experience shows that improvement projects rapidly grow large. Improvement work also struggles to show a compelling business case, even when we have aligned it to business requirements. Partly this is because a good solution is expensive, and partly because much improvement is seen as “housekeeping”: non-strategic, non-transformational.

In small-to-medium enterprises there is even less appetite for formal projects. So I suggest you (mostly) don’t generally use projects to improve services.

Where else can we source the resources to make the improvements? From “business as usual” (BAU). Improvement is normal behavior for professionals. It is part of our job to devote a certain percentage of our time to improving the systems we work with. We should all expect that things will be better next year; that we will make a difference and leave systems better than we found them. Improvement is business as usual. We need to make this commitment.

It is better to **commit most staff for a proportion of their time**, rather than a few staff dedicated fulltime to the program of work. This promotes buy-in, by involving all those affected. In a highly critical initiative, 10 percent of resource could be allocated. For most situations, 5 percent is more realistic, i.e. about 2 hours a week – on average – from everyone.

Take an agile approach. Work in cycles or "sprints." Within each sprint, create small teams of two or three people and ask them to look at one or two improvements at a time, determine what can be improved and make the improvements. The people who use the practices know best how to improve them. Empower staff to design and implement their own solutions.

By all means, manage all these assignments as a portfolio using Program Management methods if you have the resources to do so, but do not manage the individual assignments as projects. If we subjected each team to project management by breaking down the assignment into tasks, then estimating, resourcing and tracking each task, the management overhead would swamp us.

(Note: you can spin off work into projects sometimes, where you recognize that the work:

- Will create a capital asset
- Needs additional resources
- And/or is sufficiently complex that it is too risky without project management

Also, there is nothing to stop you managing other assignments using project management methods if you feel the need.)

As well as putting Program Management in place, you should also maintain a central plan and a central architecture of how your practices should hang together to ensure everyone is heading in the same direction.

As I discussed in an earlier chapter, service improvement is behavioral improvement. **People-change (also known as "culture change") is at least as important as changing practices** and technology. Better practices alone don't fix anything (just as technology on its own doesn't either). Improvements to practices only work if the behaviors of those affected change to adopt the new practices:

- Refine the practices (using best practice frameworks for reference as appropriate)
- Improve the technology to support them if necessary (often not)
- And most of all, change the behaviors of the people involved in order to adopt the new practices

A good length for a sprint is two to three months. This sounds like a long time, but it is not when considering human rates of change (as compared to technological ones). It is a short time when it comes to improving practices and changing behaviors. There is only so much that can be achieved by a small number of people devoting 5 percent of their time for a few weeks. Assignment teams must look for pragmatic outcomes that address the most urgent needs and/or the low-hanging fruit.

Some progress is better than nothing. If we try to take a formalized project-managed approach to service improvement, the outcome for the few aspects addressed by the projects will be a good complete solution… eventually, when the projects end. Unfortunately, the outcome for the many aspects of service delivery not included in the projects' scopes is likely to be nothing. Most organizations don't have enough funds, people or time to do a formal project-based improvement of every aspect of service management. We aim to address a wider scope than projects can – done less formally, less completely and less perfectly than a project would.

In many situations, "best" is overspending: "good enough" will do ("copper not gold"). And for many organizations (perhaps most), it is all very well for the experts to go on about "have to" and "must," but there are only so many resources available and we must work within the bounds imposed on us. The ideal gives us something to aim for but we should accept when we cannot achieve it.

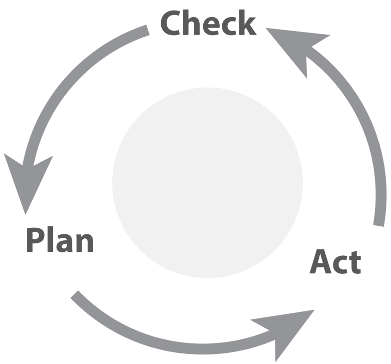
Keep going

Even though you don't have to improve forever, don't stop once you have done what you set out to do, for three reasons:

1. Anything left alone will run down. You need to keep it alive, keep it working.
2. The world changes, requirements shift. You need to continually adjust.
3. Most organizations want to keep getting better.

It's a journey that never ends. Service management transformation is not a project with an end-date. Assign it to someone and make them accountable. It is continual, not necessarily continuous: you don't have to work on it every day but you do need to keep coming back to it. Implement that formal improvement program. Get into this continual cycle: check your current state; plan where you want to be and how to get there; act to make it happen. (The most common version of this cycle is known as the Deming Cycle of Plan-Do-Check-Act, but I think Check-Plan-Act makes is more logical. I first saw it in Understanding Your Organization as a System, Vanguard Consulting, 2001, www.systemsthinking.co.uk/5-1.asp).

Too many change programs lose momentum because management moves on or loses interest. This has facetiously been described as Plan-Do-Stop. This is why you need an ongoing formal program of work to provide continuity and permanence.



The complete support desk

GoToAssist continues to simplify IT support by providing easy, affordable access to an essential toolset – service desk management, remote support and IT monitoring – all from one easy-to-use interface. Combining critical tools enables IT departments to be more efficient and effective and save costs. Customize GoToAssist to fit your unique business needs — choose one module, two or all three.

- Use GoToAssist Service Desk to log and track incidents, deliver end-user self-service and manage configuration, changes and releases.
- With GoToAssist Remote Support, you can deliver on-demand support and access unattended servers and workstations.

- Use GoToAssist Monitoring to proactively monitor your entire IT infrastructure, including critical servers and services.

Key benefits of integrated IT tools

- Gain total visibility into the entire IT support services to accelerate issue resolution
- Enhance efficiency with combined reporting and single-click functions between modules
- Deliver multiple critical IT support services from one easy-to-use interface
- Streamline workflow among team members
- Easily and quickly implement ITIL and ITSM best practices

Learn more about delivering IT services from one integrated, cloud-based toolset — visit www.gotoassist.com.

Additional resources

Rob England's "Checklists for Getting Service Management Right"

Rob England's *Basic Service Management* and *Standard+Case* (both available at Amazon and elsewhere.)

GoToAssist Service Desk Training Videos and Getting Started Guides

About the author

Rob England is an independent IT management consultant and commentator based in Wellington, New Zealand. Rob is an internationally recognized thought leader in IT service management (ITSM). He is a published author of several books and many articles and is best known for his controversial blog and alter ego, the IT Skeptic.

Additional resources

news.citrixonline.com/resources
Access upcoming and on-demand webinars, case studies and best practices white papers about working from anywhere with anyone.

www.workshifting.com
Read the blog dedicated to all things related to workshifting (getting work done outside the traditional office), including best practices tips and tricks.

About Citrix
Citrix (NASDAQ:CTXS) is the cloud company that enables mobile workstyles—empowering people to work and collaborate from anywhere, easily and securely. With market-leading solutions for mobility, desktop virtualization, cloud networking, cloud platforms, collaboration and data sharing, Citrix helps organizations achieve the speed and agility necessary to succeed in a mobile and dynamic world. Citrix products are in use at more than 260,000 organizations and by over 100 million users globally. Learn more at www.citrix.com.

7.04.13/B-89226/PDF

citrixonline.com