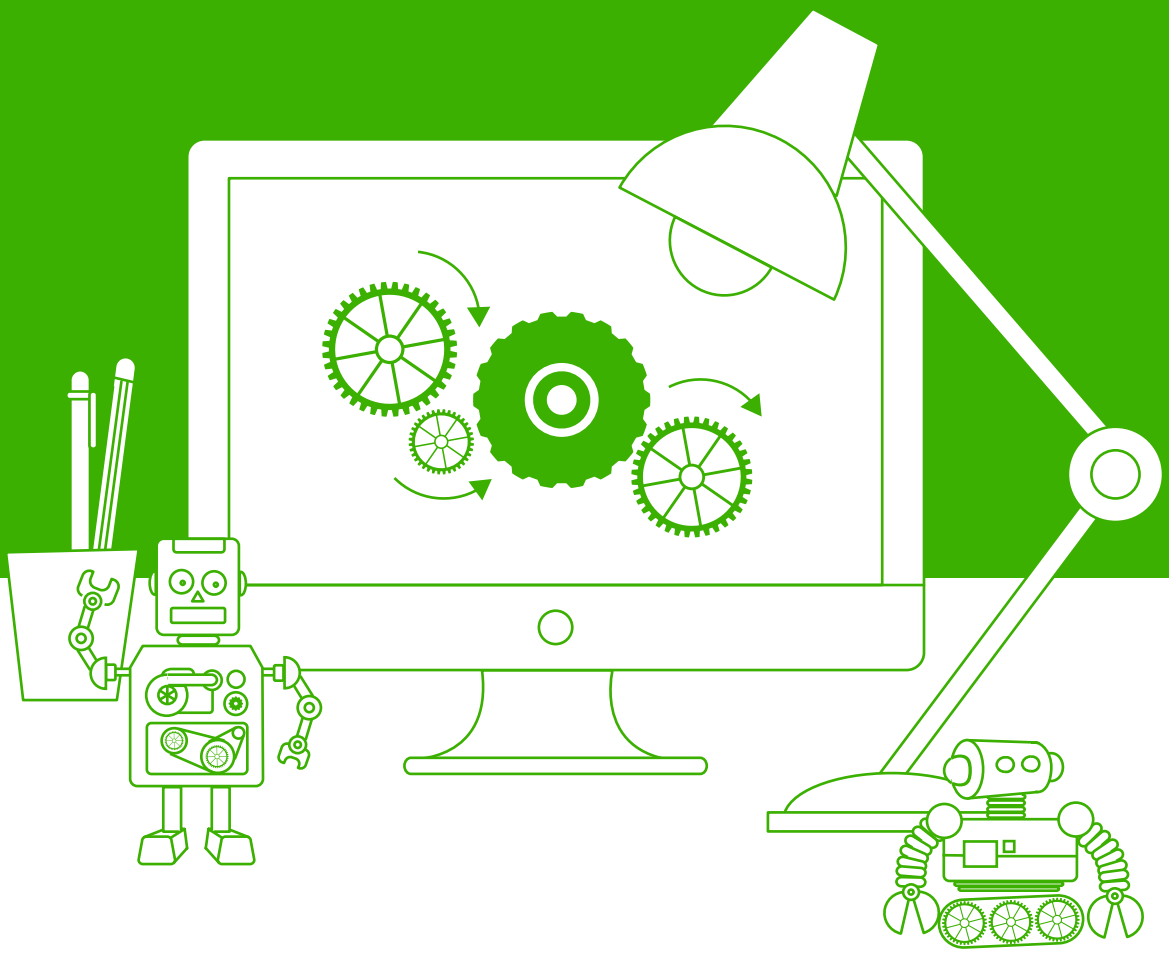
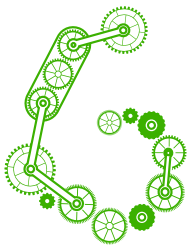
SysAid

6 Tips to Help You Improve Change Management

by Stuart Rance





Tips To Help You Improve Change Management

by Stuart Rance

Change is the single biggest cause of incidents and problems for IT services. Estimates of quite how many incidents are caused by changes vary, but I have personally seen one very large organization reduce the number of incidents by 80% for the duration of a three month change freeze. On the face of it, this might suggest that we should just avoid making changes to IT, which may well account for the resistance to change that is typical of some IT departments. But most organizations can't simply stop making changes. A 'no change' policy would have an even bigger negative effect on customers than the change-related failures it was intended to prevent.

What is a change?

ITIL (the best practice framework for IT service management) defines a change as "The addition, modification or removal of anything that could have an effect on IT services…"

This is where change management comes in. IT organizations implement change management to try and protect the services they deliver to their customers. They put procedures in place to ensure that changes are well documented, that they have been tested, that the risks have been understood and that suitable plans are in place to deal with things that might go wrong.

Ideally, change management should protect your IT services from the impact of change without having any negative consequences, but in practice that is not always the case. All too often it becomes over-bureaucratic. There may be complex forms to fill in, and a seemingly endless series of reviews and checks to complete, resulting in painfully long delays before even the simplest change can be implemented. Many IT organizations have regular Change Advisory Board (CAB) meetings, where changes are discussed, and decisions are made about which changes will be implemented. These CAB meetings typically take place once a week and, at their worst, can involve lots of people with formal change management responsibility sitting round a table discussing changes that they know very little about while lots of people with change requests are obliged to sit and wait till their particular request reaches the top of the agenda.

If this sounds familiar then read on to discover some of the things that my customers do to get real value out of change management. There are things you can do to provide the protection your customers need without creating an unwieldy process that slows everything down; you just need to focus your efforts on the things that are really important to you. A good change management process can really help to reduce the number of incidents, whilst at the same time it can facilitate required changes and improve the availability of IT services.

What Is the Purpose of Change Management?

I once heard change management compared to the brakes on a car. If you own a car with very poor brakes then you have to drive very slowly, because you don't have the level of control required to stop when you need to. If you want to drive fast you need good brakes.

If you allow uncontrolled changes, with the high failure rates that follow in their wake, you put both your organization and your infrastructure under intolerable pressure. A good change management process increases the percentage of successful changes, decreases the number of changes that need to be backed out, and decreases the impact of failed changes. Put this in place and you can accommodate – perhaps even welcome – a much greater rate of change. Like good brakes, good change management gives you the control you need to make more changes and to implement them faster.

Getting the Balance Right

Your change management process should have two clear goals:

- To facilitate the rate of change needed by customers to help them meet their business goals
- To reduce the negative impact of change on both IT and customers, by managing the risks that change can create for IT services, as well as for underlying applications and infrastructure

There is an obvious tension between these two change management goals. If change management is too risk averse, it can stifle business innovation. However, if changes are implemented with insufficient care, then poorly managed risks can lead to losses, which may have an even bigger impact.

Issues will often arise when an organization focuses too much on one of the two change management goals at the expense of the other. It is very important to get the balance between them right. The balance you need will, of course, depend on the nature of your organization, and of the IT services you are delivering to your customers.

Here are some tips to help you make sure your change management process is effective.

1 Categorize IT Services Based On Their Risk Appetite and Need for Agility

Some IT services need very high levels of confidentiality, integrity, and availability (CIA). These services are changed very infrequently and they typically need a very formal approach to change management, to help ensure that they meet their CIA requirements. They are often subject to legal or regulatory requirements as well, so that all changes need to be fully auditable.

Other IT services may need slightly lower levels of CIA, but a much higher level of agility, to support rapidly changing business needs and customer expectations. These services are often directly visible to end customers, and may include web-based access, or apps that are downloaded to tablets or phones.

In between these extremes you may have services that call for a balance between agility and CIA.

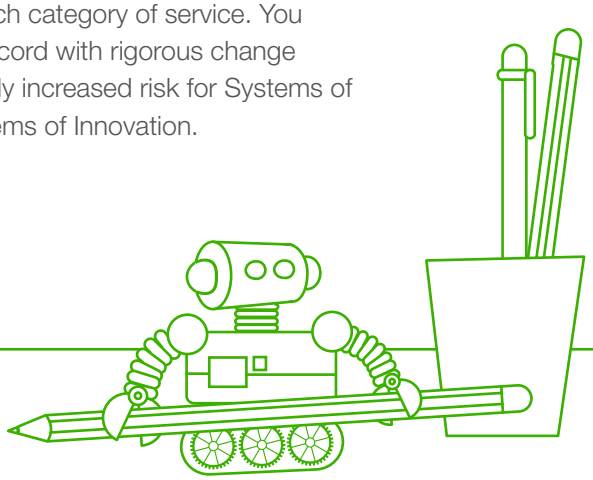
It is not appropriate to manage changes for all of these services in the same way. You should create a set of well-defined categories, and assign each service to a category. For example, Gartner has defined three common categories for IT systems.

- **Systems of Record:** These support an organization's critical data and transactions. They typically have a low rate of change and may be subject to regulatory requirements.
- **Systems of Differentiation:** These support unique company or industry capabilities. They need fairly frequent reconfigurations to meet evolving business or customer needs.
- **Systems of Innovation:** These are often built or modified on an ad-hoc basis to meet new requirements or opportunities.

You may want to adopt this terminology, or to create categories of your own.

Once you have defined the categories you should talk to your customers about them. You need to reach an agreement about which category applies to each service. This helps your customers to think about their differing needs for agility and CIA, and then express their needs in a way that is easy for you to understand and document.

You should then design your change management process to deliver the level of risk management and agility your customers require for each category of service. You will need to ensure that you protect the Systems of Record with rigorous change management, while allowing faster changes with slightly increased risk for Systems of Differentiation, and a very high rate of change for Systems of Innovation.



2 Use Standard Changes Wherever Possible to Reduce the Need for Change Approval

Allowing staff to implement changes that have not gone through a change approval process can be risky, but requiring change approval for absolutely everything can be slow and frustrating. One solution to this dilemma is *standard changes*.

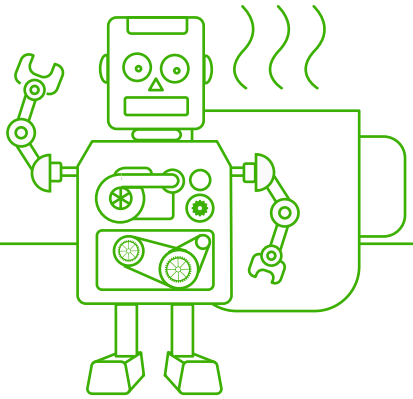
A standard change is a pre-authorized change that is low risk, relatively common, and follows a well-defined procedure or work instruction. The great thing about standard changes is that you can test and approve the procedure or work instruction and then allow many changes to take place without the need for any further testing or approval.

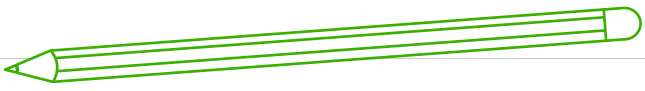
You should define standard changes for as many situations as you can. Good candidates are changes that you need to implement quite often, where you can define the exact steps needed, and where the level of risk involved is acceptable to you. When you have identified a good candidate, get your best people to design the procedure for carrying out the change, get them to test it, and get it approved. The procedure can then be reused without further approval whenever appropriate, and without undue risk.

Typical examples of standard changes are installations of common hardware and software, password resets, or deletion of temporary files to free up disk space. You may want to define many different standard changes for your Systems of Innovation, while retaining greater control over your Systems of Record by using standard changes less often.

Difficulties with standard changes are unlikely, but nothing ever works perfectly in all circumstances, so you need to have a procedure in place to manage this too. If anybody identifies an issue with the procedure for a standard change, get them to submit a change request to have the standard change modified. This ensures that everyone can learn from each other's experience.

Most IT organizations that I have worked with have far too few standard changes. This is a great pity. Standard changes can be a very effective way to deliver both good risk management and high levels of agility. Why not carry out a review of your last 12 months' change requests to identify opportunities for creating new standard changes?



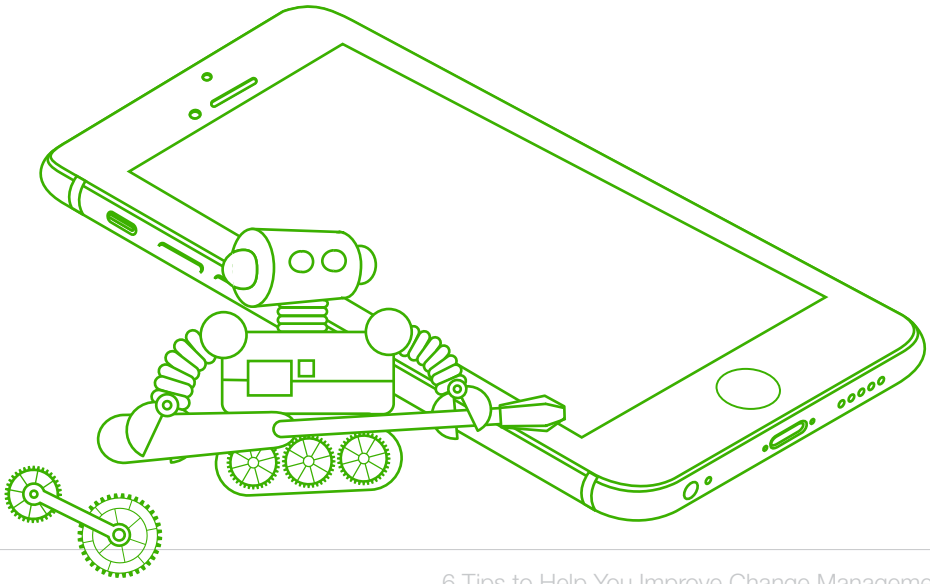


3 Define Change Models to Reduce the Amount of Effort Needed for Each Change

A *change model* is a predefined set of activities that helps to reduce the risks associated with carrying out a particular type of change. Examples of change models are the activities required to open a port on a firewall or to deploy a patch.

Unlike a standard change, a change model is not pre-approved. The detailed deployment activities may need to be different each time the model is used, so it may also provide fewer detailed instructions about the precise steps to be followed. Nevertheless, change models can help to reduce risk by ensuring that key steps are not forgotten when carrying out a change. For example, a change model for upgrading the operating system on a server may specify who needs to be consulted to approve the change, what steps should be taken to migrate the service to an alternative server, and what application testing is required. This leaves the decision about whether (and when) to upgrade a specific server to the people who are best placed to make this decision, whilst helping to ensure that, whenever they are implemented, server upgrades are successful.

Like standard changes, most IT organizations have far too few change models. Why not carry out a review of your last 12 months' change requests to identify opportunities for creating new change models?



4 Integrate Change Management with Your Software Development Lifecycle

We think of change management as a separate process, since it has its own activities and tools. But it can be much more helpful to think of it in the context of how your organization creates value. Change management does not, by itself, create any value. It only creates value as part of a whole value stream that starts by identifying the requirements for a new or changed IT service and ends with the changed service being available for use.You can find some more thoughts about this idea of value streams in the IT4IT Reference Architecture.

In a traditional IT department, the operations team runs change management. It is there to protect operational services from the effects of software that originates somewhere else. It may have come from a development team or a project. When the development team, or project, gets to a particular stage it has to submit a change request and wait for approval from the operational team before proceeding. This introduces a bottleneck into the workflow, potentially creating significant (and frustrating) delays at a critical point for the team requesting the change. This can be a very inefficient way of working.

To get round this bottleneck, some organizations have started to work in a much more integrated way. The approach taken in a DevOps environment is to have a single team that owns the entire lifecycle, from requirements right through to operations.

DevOps also includes a number of other ideas that are very significant for change management:

- **Continuous integration** merges all new code into a single stream. It does this very frequently, and with automated testing. Consequently, there is no need to run multiple separate development streams and no need for complex integration at a later stage. This can result in simplified releases.
- **Continuous deployment** automatically deploys tested code into production environments. Since each change that is introduced is very small and the process of introducing it is fully automated, the risk of any errors being introduced is very small too. This can enable very high rates of production change with minimal risk.
- **Infrastructure-as-code** uses an automated process to create operating environments in the cloud. This ensures that the correct infrastructure is in place every time, and that it has been created accurately and reliably, so you can be confident that production and testing will always run in the same well-defined environments. It also supports configuration management by deploying the infrastructure from a source controlled data file.

These DevOps ideas tend to work very well for Systems of Innovation, but many organizations are reluctant to use them for their Systems of Record. It is possible, however, to adopt some ideas from DevOps even when dealing with the most critical systems with very high regulatory requirements.

The most critical DevOps idea to adopt is to ensure that your processes are all designed together, with an understanding of how they will deliver value to your customers. When this is done well, it can result in significant improvements in change management throughput and reduction in change management effort. For example, one organization I worked with had a single process owner who designed the entire "requirements-to-deployment" value chain. This organization had a fairly traditional organization design, with separate teams for requirements engineering, software development, and IT operations. The new process owner had to work with the managers of all these teams to create an end-to-end process (and tool) design that met all their requirements. Once this new process was in place, it ensured that all the activities flowed seamlessly into each other. When the developers were ready to deploy newly developed code, they simply had to click a button in the user interface of the service management tool set. At this stage everything needed to submit the change request was already in place, and the tool was able to create the change request with almost zero effort from the developers. Similarly, the change manager was able to verify that all required testing had taken place with no effort, because the tool design and the process flow had already verified this information.

It does take some effort to redesign your processes so that they all fit together seamlessly, but the benefits can be very significant. Improvements you make in the workflow apply to every service you support, whether it's a highly agile app-based service or a highly regulated legacy financial application.

There are management tools and ideas that can help you to manage workflow, including Theory of Constraints and Kanban. Detailed discussion of these would make this paper far too long to be readable, but further study of both these areas is worthwhile.

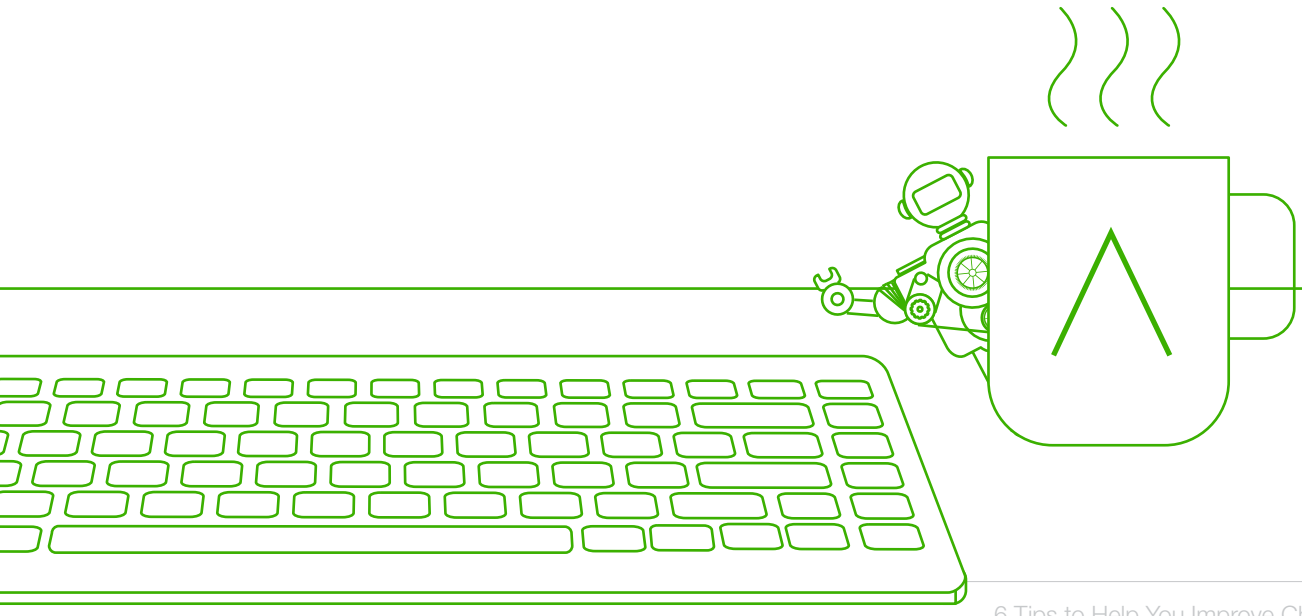
5 Minimize the Number of Approvers Needed for Each Change

There is an old saying that if everybody is responsible for something, then nobody is responsible. This applies particularly to approval of change requests. I have seen organizations where there are 20 or more people in a CAB meeting, and where anyone can reject a change, but nobody is authorized to approve it. When the change later fails, nobody takes responsibility for understanding why the wrong decision was made, and for improving things for the future. It would be much better in these cases to involve far fewer people in making the decision.

In one organization I worked with, each change request had exactly one approver. Information provided in the change request automatically channelled the request to the appropriate change approver, who was then responsible for consulting with other people as required to ensure they made a good decision.

What made this approach really effective was measurement and reporting. Every change was reviewed to ensure it had succeeded. This was more than a quick check to ensure that the change didn't cause an incident; the change closure codes included all of the possible failure modes the organization had previously identified. (See Tip 6 below: "Measure and Report the Right Things to Drive the Behaviour You Want to Encourage".) Each change approver (and each change requestor) received regular reports showing what percentage of their changes fell into each of the possible failure modes.

The effect of this was that people really cared about the changes they approved. They took great care to review changes and ensure that they would succeed before approving them, but without the bureaucratic overhead of a meeting with 20 people. Typically the change approver would contact a range of people with different expertise to ask for their input before making a decision, but would then take responsibility for the final decision. Another advantage of this approach was that it allowed the change approver to consult a wider range of stakeholders than attend a typical CAB meeting, helping to ensure that a balanced decision was made by taking into account the views of customers and users, as well as IT staff and management.

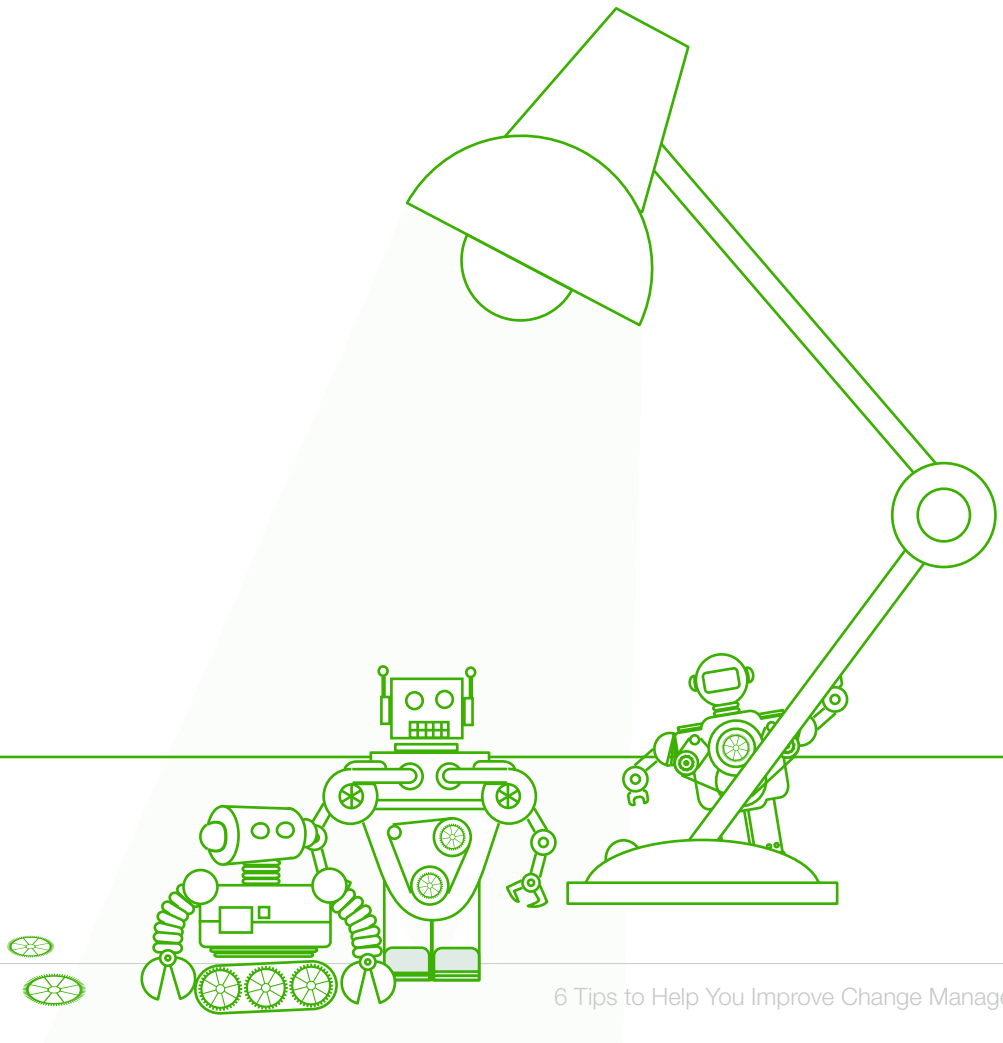


Minimizing the number of approvers needed for each change works very well when combined with the suggestion in Tip 1 - "Categorize IT services based on their risk appetite and need for agility". You could, for example, allow people to approve their own change requests, provided that:

- The change does not affect a System of Record
- The change is happening in a pre-approved timeslot
- The change requestor rates the risk as low
- The change requestor has not approved any changes that failed in the previous 12 months

If people are measured on their change success rate, and are no longer allowed to approve their own changes for a period of time after a failed change, then they will be very careful only to approve changes where they have a high level of confidence.

If you have many people involved in regular CAB meetings, where few of them are making a significant contribution to each decision, then you should consider whether you can improve efficiency by reducing the number of people involved, but making those people accountable for the decisions they make. You may not be able to go all the way to just having a single approver for each change, but you should aim to get as close to this as possible.



6 Measure and Report the Right Things to Drive the Behaviour You Want to Encourage

Metrics can have an enormous impact on how your organization performs. If you measure and report the wrong things, then you will drive the wrong behaviour in your staff, and you won't get the results you want for your customers.

Earlier in this paper I noted that change management should have two clear goals:

- To facilitate the rate of change needed by customers to help them meet their business goals
- To reduce the negative impact on both IT and customers caused by change

Many IT organizations measure and report the number and percentage of failed changes as their key change management metric. If you only measure and report the negative impact of change, then you will create a culture that resists change, and fails to meet your customers' needs for business agility. You need to create metrics that support both of these goals, and maintain the tension between them to help you achieve the right balance.

The first thing you need is a way to evaluate the success or failure of each change properly. The concept of a "failed" change is too vague to capture all the different ways that changes can go wrong, and often results in changes being reported as successful even when they completely fail to deliver the business benefit the customer expected. At a minimum, every change should be evaluated against the following criteria:

- Was the change deployed to all intended locations within the expected time and budget?
- Was the change fully deployed with no requirement to roll back any part of it?
- Did the change cause any incidents, either to the new or changed service, or to any other IT service?
- Did the change deliver all of the expected business benefits?

A change is only completely successful if you can answer yes to all of these questions.

Once you have established this approach to measuring change success, you can review your critical success factors (CSFs) and key performance indicators (KPIs) and define these in a way that will encourage the behaviour you want.

I have written about defining CSFs and KPIs for change management before, so I won't provide more detail here, except to say that you must have a balanced set of metrics that encourage your IT staff to deliver a balanced approach to change management. Getting this balance between managing risk and achieving business agility is not easy, but recognizing that you need to do it is an absolutely essential first step.

Summary

The tips in this paper are intended to help you improve how you manage IT changes, getting the right balance between reducing risk and facilitating the rate of change that your customers need:

1

Categorize IT services based on their risk appetite and need for agility

Different IT services have different requirements for confidentiality, integrity, availability and agility. You should create a standard set of categories to enable you and your customers to understand the required balance for each service.

2

Use standard changes wherever possible, to reduce the need for change approval

Standard changes enable you to test a change once and then deploy it many times. This can lead to reduced risk and increased change throughput. Most IT organizations have far too few standard changes.

3

Define change models to reduce the amount of effort needed for each change

A change model has less detail than a standard change, and is not pre-approved. It can help to reduce risk by identifying the key activities required for a particular type of change.

4

Integrate change management with your software development lifecycle

Change management only creates value as part of a value chain that starts with defining requirements and ends with deployment of a new or changed IT service. The tools and processes for this whole value chain should be designed together to reduce bottlenecks and provide the greatest possible efficiency.

5

Minimize the number of approvers needed for each change

If there are too many change approvers then nobody is responsible for making the decision. The ideal situation is to have just one approver for each change, and for this person to consult with others, as necessary, to ensure they make the right decision.

6

Measure and report the right things to drive the behaviour you want to encourage

Every change should be evaluated to ensure that it was fully deployed to all agreed locations, that it was installed within the agreed time and budget, that it didn't cause any incidents, and that it met the customers' expectations. Change reporting should be designed to ensure that it drives the behaviour you want, in terms of both risk reduction and facilitation of business agility.

If you follow these tips, then you may find that your customers start to treat change management as something useful, because it helps to create business value, rather than as a necessary evil that they put up with because it protects them from risk.