invgate

ITSM101

Change Management Tips

By InvGate, January 2018

Introduction

Most, if not all, IT service management (ITSM) professionals know that the change management process ensures that changes are delivered swiftly; risks are effectively managed, with changes reviewed, approved, and scheduled in a sensible way; that regulatory requirements are met; and lessons learned are captured. So why do so many still struggle to get the basics of change management right?

ITIL – the popular ITSM best practice framework – states that the objective of change management is:

> *"To ensure that changes are recorded, evaluated, authorized, prioritized, planned, tested, implemented, documented, and reviewed in a controlled manner."*

In other words, that change management is the thing that makes sure we don't break anything when introducing, moving, or retiring IT services – which sound eminently sensible.

Thus, when describing the change process, the people involved – from change managers to change advisory board (CAB) members – are the guardians or protectors of our live production environment and business operations, as well as the facilitators of change.

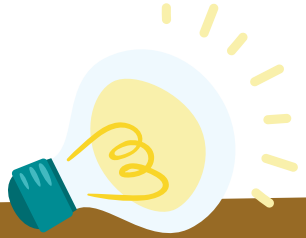
Ultimately, change management is in place to protect everyone, in this case by ensuring that all changes are sanity checked, tested, reviewed, authorized, communicated, and scheduled at a sensible time. With the key change management areas being:

- Creating the change
- Reviewing and evaluating the change
- Change assessment
- Authorizing change build and testing
- Coordinating change build and testing
- Authorizing change deployment
- Coordinating change deployment
- Review and closure.

Much of how to do the above is included in ITSM best practice documentation such as ITIL. So, this paper instead drops into the detail of change management to offer tips that will help your organization at a more granular level, through six pieces of advice:

1. Making your change request form easy to use
2. Making your initial “sanity check” count
3. Supercharging your CAB
4. Managing change build and test better
5. Implementing change better
6. Making your change reviews count.

Please read on for more on each of these tip areas.



1. Making Your Change Request Form Easy to Use

One of the golden rules of ITSM is to always try to make it easy for people to use your process.

If raising a request for change (RFC) is a 45-minute saga of multiple forms, menus, free-text fields, and radio buttons then people will try to avoid it either by not engaging with the process (leading to unauthorized changes) or by using the standard-change process (a quick route for pre-authorized, low-risk changes) inappropriately.

So, make your RFC form as easy as possible to use, such that RFCs can be quickly raised in an accurate, consistent way. The form should be designed to elicit the important information needed to assess a change, not everything the requester knows on a particular topic. While less isn't necessarily more, it's important to have the right information in change requests for review and approval purposes. Those involved need to have all the facts required to make the right decision for your end users/customers and the business.

Your change form should thus include information related to the following:

- Title/description/reason
- Service(s) affected
- Whether the configuration management database (CMDB) – if you have one – needs to be updated afterwards
- Information security considerations to be taken into account
- Known risks
- Implementation windows
- Implementation teams
- Pre-implementation testing
- Implementation plan
- Post-implementation verification checks
- Rollback plans
- The impact on other environments
- Whether the change needs be replicated to your disaster recovery (DR) environment.

If all of this is required, make it easy for the information to be submitted.

2. Making Your Initial "Sanity Check" Count

The review and evaluation of a change request should be the initial check carried out by the change manager. This is to ensure that the change is reasonable and has all the required information filled in. No one wants to attend a CAB meeting only to find that half the change requests on the agenda either have missing information or haven't been raised correctly. So, putting the work in early will pay off later.

Things to bear in mind when assessing a change request include the benefits, hopefully putting the needs of the business first and foremost. Because you can have all the technical benefits in the world but you'll need to be able to articulate them in business language such that the change can be reviewed and communicated effectively across CAB members.

Then there are the risks associated with the change request. It's best to ask as many "potential downside" questions as possible up front so that, in the event of a change going terribly wrong, you can be confident that everything possible was done to mitigate risks and to protect services, and business operations, during the change process.

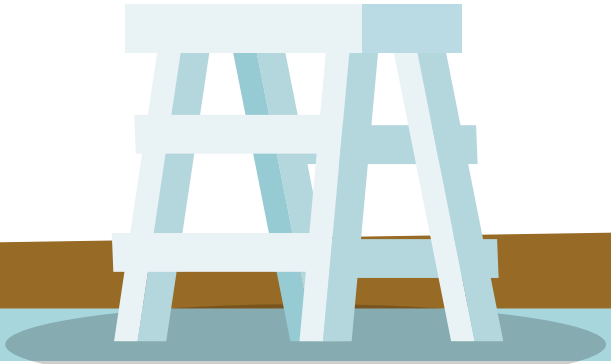
Things to consider when carrying out a change risk-analysis include:

- The number of people affected
- Whether the change needs to be done out of hours
- Financial, regulatory, and reputational aspects
- The effect on service performance – is there any potential loss of productivity or hit on performance?
- Availability – is there any potential down time?
- Seasonal considerations, for example is the work being carried out too close to a business-critical time?

Also, make sure that you have clear assessment criteria for the management of changes. Examples include:

- **Pre-implementation testing** – how do we know this change will go smoothly? What tests are planned so that the best possible outcome can be achieved?

- **Implementation plan** – does it make sense, do we have the right people involved, if other teams are needed to support the change are they aware and do we have contact details for them? Are there any areas where we might need to conduct checkpoint calls or bring in additional support to reduce the risk?
- **Communications** – will there be any downtime? Do we have a way of letting the business know? Do we know who we should be telling?
- **Back out plan** – what happens if something goes wrong during deployment? Do we "fix on fail" or roll back? Are the change implementers empowered to make a decision, or is escalation needed?
- **Post-implementation verification** – how do we make sure that everything is as it should be?
- **Impact to other environments** – ensure that any pre-production environments will be refreshed in line with production following the change. More importantly, make sure that if the change is applied to your live environment, it's also applied to your disaster recovery (DR) or continuity environment. The last thing you need in the event of a disaster is not being able to fail over because someone forgot to enact this key step in the change process.



3. Supercharging Your CAB

The CAB meeting is one of the most important and useful meetings a service-orientated organization can have. It sets out a view of what's happening to key services over the next week or month, reviews previous change activity, and looks at continual service improvement (CSI).

So, when setting up your CAB, make sure that you have a clear "terms of reference" statement that gives attendees the right steer on how to prepare, good meeting behaviors, and how to represent changes effectively.

The change manager should also send out the CAB agenda, including the changes to be discussed, the change schedule, and any previous changes that caused incidents, well in advance of the CAB meeting. Why? Because those involved need sufficient time to read and consider the changes – which might call in third-party knowledge and opinions – as well as to identify any potential issues or questions.

Of course, not every change has to go to the CAB. Just use the CAB for the important stuff – your big, complicated changes that will have a major impact on the business. So, scheduled server reboots or security patches are candidates for a standard-change model – keep the CAB as focused as possible by enforcing an agenda that only deals with the RFCs that are high risk, major impacting, or have lots of complicated detail. And ensure that the right people turn up, with all required business areas represented.

An effective CAB agenda looks something like:

- Review and retrospective approval, where appropriate, of any emergency changes
- Review of implemented changes
- Review of incidents caused by change, including any lessons learned
- Review the forward schedule of change
- Review candidates for new pre-approved, standard change models
- Improvements and opportunities for CSI.

Finally, it's important to keep your CAB pacey, and on schedule, so that it becomes seen as a "must attend" event and a business enabler, rather than yet another mandatory meeting to be sat through.

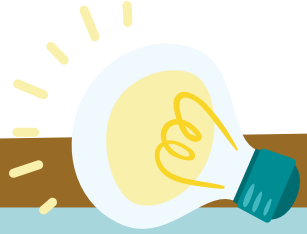
4. Managing Change Build and Test Better

One of the biggest factors in increasing the probability of a change being deployed successfully is ensuring that the appropriate levels of testing are carried out – it helps to make sure that the expected outcomes match reality.

Testing the change also gives you the opportunity to work out any issues with the deployment procedures you've created. And one of the first things to look at, while also making life easier for change testing, is the development of standard build and test methods. And if you can use automation, fantastic – it saves time, removes duplication, reduces the likelihood of human error, and the economies of scale can be significant.

If automation isn't an option, ensure that your build and test methods are documented and templated where possible. Have a list of standard tests and checks to follow, so that even if you can't remove the potential for error completely, you can ensure that the testing is being carried out consistently and that nothing is missed.

Also ensure that your build and test environments are fit for purpose. For instance, does the environment match the type of testing to be carried out?



5. Implementing Change Better

After all the pre-work, the end is finally in sight. The change has been raised, sanity checked, reviewed, authorized, and tested. Now it's time for go-live. But before people can sit back and relax, they need to survive the implementation stage.

Great communication is the key to a successful deployment. So, ensure that if downtime is involved, business approval has been obtained, a communications plan is in place, and the appropriate service level agreements (SLAs) have been relaxed.

Also, ensure that the agreed implementation plan is followed – yes, this needs to be stated – and that everyone involved has the contact details (and back up contact details) for the other invested parties. For example, there's nothing worse than being involved in a change in the middle of the night that's overrunning because you can't track down the person who's deploying the code, bouncing the server, or rerouting network traffic.

Finally, once the change has been made, make sure that the post-implementation checks are carried out – is everything as it should be, and are there definitely no adverse effects? If the change has not been successful, then the remediation plan discussed and agreed at the CAB should be followed.

6. Making Your Change Reviews Count

The change has been implemented and everything has gone to plan, or at the very least nothing is on fire. The final step is to carry out a change review, to look at both what went well and what could be improved on for next time.

Let's start with the positives. If the change went well, then that's great. Look at what was done and who was involved so all the great work can be captured for next time. Perhaps in the form of change models or standard changes.

However, if things went badly. For example, if the change broke something, caused an incident, exceeded the implementation window, or had unforeseen effects, a review needs to be carried out to understand what happened, the root cause, any remedial actions, and finally, any actions to prevent a recurrence.

If the change has caused incidents, then the change manager needs to be involved with both the incident and problem management processes. They'll be providing information to incident management about the nature of the change and exactly what was done during the fix efforts and, later on, they'll be looking at lessons learned with problem management in an effort to prevent recurrence. Above all, lessons learned need to be captured, discussed, and acted upon. And a great way to make sure this happens is to have a regular CAB agenda item for lessons learned and CSI.

When reviewing changes, it's important to look at what happened from a business perspective. So, speak to your customers and ask them if they feel the expected results were delivered and that the promised benefits have actually materialized. The next step is to look at the technical benefits. For example, are we in line with the current recommended levels of patching, has performance been improved, or do we have more resilience?

The final stage of the change review is to close the loop. Making sure that infrastructure and IT service records – a CMDB or similar – are updated such that what's captured in documentation matches what's in the production environment. And if the change was to fix a problem record, then work with problem management to ensure that the issue has been resolved and with the service desk to let everyone else – including those affected – know.

Summary

Effective change management can deliver a wealth of benefits – with changes delivered quickly but in a controlled manner, change risk effectively managed, change delays and failures reduced, and change-related incidents and problems minimized.

And change doesn't have to be a slow, bureaucratic process – particularly when the CAB is involved – if sufficient thought and effort is invested in the change management process and practices. Ultimately, as with any ITSM process or capability, change management will only ever be as good as you make it. So think long and hard about how to make change management an easy-to-use facilitator rather than a brick wall to be avoided at all costs.

About InvGate

InvGate is a provider of IT service management and IT Asset Management solutions, designed to simplify and improve the lives of IT professionals.

InvGate Service Desk helps customers to provide better IT support, offering a single point of contact for end users to report IT issues and make requests for new services. With capabilities aligned with the ITIL best practice framework, InvGate Service Desk enables your company to improve IT support efficiency, to reduce costs, and to improve the quality of service and the customer experience for end users.

If you'd like to try InvGate for yourself, then you can start your free 30-day trial today.