ITIL®

Service acceptance criteria

Sometimes you got to be cruel to be kind

Julia Harrison

AXELOS.com

AXELOS GLOBAL BEST PRACTICE

Practical Guidance
June 2017

Contents

1 Service acceptance criteria

There is much talk in ITSM circles about the benefits of introducing DevOps work practices into service management. In particular, the idea of self-organizing, cross-functional teams who release software hundreds of times a day is certainly appealing. When done well, DevOps can enhance the service that the organization provides for its customers.

But what about organizations who have not yet built out their automation capabilities or, more importantly, have not put in place the cultural changes required to make the introduction of DevOps viable? This is a challenge faced by many environments with larger, less frequent changes to their software.

The goal of this article is to use the ITIL® Practitioner guiding principles "start where you are", "collaborate" and "keep it simple", to show how it is possible for IT service operations within these organizations to reach across the divide to software development in order to bring improvements to their services.

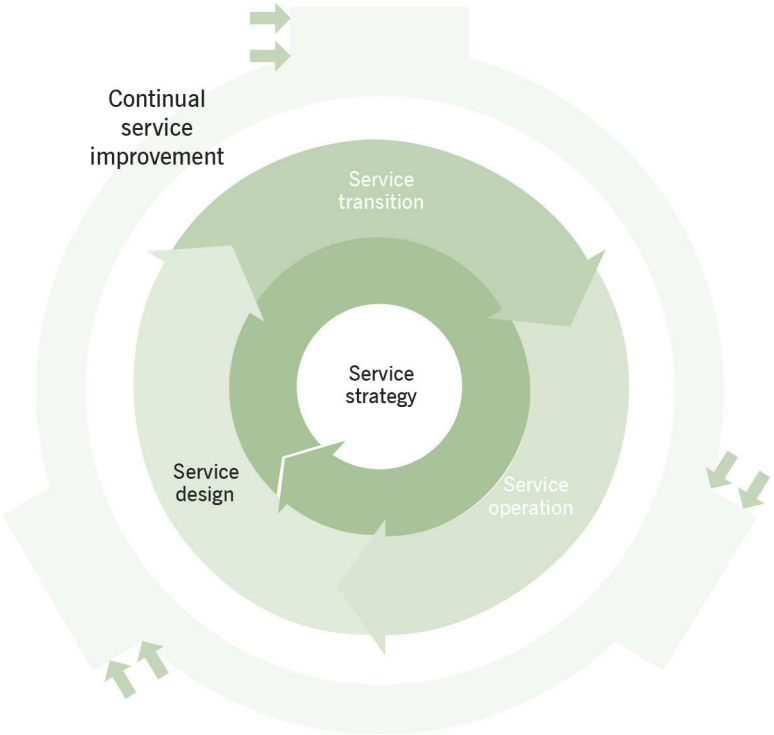

Figure 1.1 The ITIL service lifecycle

If you are trained in ITIL, you are probably familiar with the service lifecycle figure (Figure 1.1) and its significance as a cycle. There is a circular flow from service design to service transition to service operation and back to service design. Everything is aligned with service strategy and the culture and practices of continual service improvement.

1.1 STUFF FLOWS DOWNHILL

For many people at the sharp end of IT service, this harmonious feedback loop is a distant dream. Lots of organizations run their IT services with something that looks more like a waterfall. A cascade of 'stuff', if you will. And, as anyone who has spent time working in a service operations function ('ops') will tell you, this 'stuff' flows downhill.

A typical ops team spends much of its time dealing with the output of the service transition processes, the new and changed services within the production environment. Service transition, in theory at least, can only make something happen if the service was part of service design. If those lifecycle stages are working optimally, ops will be involved from the early stages of any change project, and releases will come with documentation, support tools, training, and everything that is needed to successfully support the change.

In less ideal circumstances, there is a feeling of having been thrown a hot potato.

Service acceptance criteria (SAC):

A set of criteria used to ensure that an IT service meets its functionality and quality requirements and that the IT service provider is ready to operate the new IT service when it has been deployed.

Acceptance:

Formal agreement that an IT service, process, plan or other deliverable is complete, accurate, reliable and meets its specified requirements. Acceptance is usually preceded by change evaluation or testing and is often required before proceeding to the next stage of a project or process.

Example:

In the case of the deployment of a new software package, service acceptance criteria would typically include:

- checking that the software meets business requirements
- testing to ensure that the software works to the required quality and performance
- making sure that any training, documentation, tools, etc. required to support and maintain the software have been provided to the support organization

Ideally, these are specified ahead of time as part of the service design package, and are checked and signed off by the service validation and testing process. Organizations where service design and service transition processes are less mature have the most to gain by starting to use service acceptance criteria.

1.2 COMMUNICATE, EARLY AND OFTEN

So what do you do if you find yourself in this less than ideal situation? One obvious improvement is to establish better communications and collaboration earlier in the lifecycle. If ops can be consulted when service design begins, there is an opportunity to work out the service acceptance criteria (SAC) as early as possible, which means the deliverables for the SAC can be built into the project plan and tracked by project management. It does not matter that the details of the design are not yet known or signed off. At this stage, 'good enough for planning' is fine. Regular contact between the various teams will allow lightweight, provisional SAC to be developed where there is uncertainty; the gaps and clarifications can be dealt with as the project progresses.

In my experience, most project managers are happy to collaborate if it means turning the amorphous blob of 'handover' at the end of a project plan into a list of tangible deliverables they can track.

Developers, engineers, service designers and others currently on the 'throw' side of the hot potato analogy may take some convincing to give ops a greater role in service design. However they want the same thing as ops on the 'catch' side: for their hard work to result in services that operate successfully and create value for the business. To initiate an improvement, we need to make sure that both sides understand that they are working towards the same goal, and show them that sharing information early and often will help achieve it.

Realistically though, talking about a shared purpose is not always enough to inspire action. If you are in the ops team and struggling to get other parts of the organization to share your sense of urgency to improve, you may have to lead the way.

1.3 LEADING THE WAY

Firstly, if you don't already have one, create a template for service acceptance criteria. This will be a 'shopping list' of all the items you might need for any successful handover. Most handovers will not require all of the items on the list, and all of them will require you, the ops team, to provide the detail to make the items specific enough to work on. Having a template makes life easier later on, and helps set expectations about the kind of things you will need.

Example:

A template item might be "Familiarize Service Desk with changes". For a minor version upgrade, that familiarization might take the form of a link to the user release notes. For a whole new service, it might require in-person training sessions for the Service Desk agents. It is up to the ops teams to list what is necessary and appropriate in the service acceptance criteria for that specific release.

Share your template with the developers, engineers, service designers, and anyone else who needs to be aware of it. A one-off slot in a regular team meeting could be a good opportunity to explain the template and solicit feedback. Put a copy of the template in a place where others can refer to it for a general idea of the kind of items you might need. But make it absolutely clear that the SAC for any handover has to come from ops, and will be specific to that handover. Also, let people know that you can start working on the SAC, and will share it as a work in progress, as soon as you have some information about what is being designed. So it is in everyone's interest to bring you to the table as early as possible.

Next, get a senior manager to agree that services should not be handed over without the service acceptance criteria having been met. With some luck you have this policy already, but if you find that you routinely have to break this rule then it is worth getting it reaffirmed, with a commitment to improve. You should go into this conversation armed with data and examples of how incomplete handover has hurt the business in the past. (Examples are more memorable. Data will show they are representative and not just atypical occurrences.)

Notice that I said services *should* not be handed over without SAC being met, not *must* not. Realistically, if ops are seen to be blocking everything that comes their way, any progress will quickly be undone. An exception-handling mechanism will be especially important in the early days of any improvement. Compromise will be necessary, and it is important to model the cooperation and pragmatism you want to see in return.

Example of exception process:

- Approval required from a senior manager
- Ops to agree minimum subset of SAC required for Day One
- Additional Early Life Support to cover gaps in SAC (e.g. duty rota of developers to help with supporting undocumented components)
- Agreed plan for delivering missing items from SAC
- Numbers of exceptions to be logged, reported on at an agreed interval (e.g. monthly), and monitored for improvement.

Hopefully this should be enough. In a world where everyone listens in meetings, reads all their emails and acts in the organization's best interests at all times, ops will now be included early in all new change projects.

1.4 IF EVERYTHING ELSE FAILS...

However, even when there is a willingness to take the leap, even assuming your communications were received and understood, teams working under pressure find it especially difficult to improve how they work. In stressful situations, we tend to fall back to 'muscle memory': doing things how they've always been done. If you are in an ops team in one of these unlucky situations, does that mean all is lost?

Not quite. There is a way that you can still drive the improvements, if you have the stomach for it, and if you have the support of that senior manager we talked about. To be clear: ideally, this is not how things should be done. But if all else fails, it can work. And you only have to do one thing.

Push back. Just once.

This is the likely scenario: someone will come to ops with a changed service to be handed over. Ops' response will be to learn about the change and come up with a list of service acceptance criteria, which they can do quickly because of their excellent SAC template (see above). Someone will review the list and say "we can't deliver all those things by our go-live deadline of next Friday" or words to that effect. So the developers and engineers have a difficult choice: move their go-live date or get approval for an exception. (And because you all want the same outcome, including a good relationship in the future, your side of this conversation will be supportive and sympathetic and completely free of finger-pointing and blame, obviously). If the change is urgent enough that the date can't be moved, an exception will be granted. Ops will get their minimal service acceptance criteria met, the changed service will go live, the exception plan will be put into place for filling in the gaps. It will not be perfect, but you will muddle through and it will still be better than no handover.

But most importantly, the developers, engineers, service designers, project managers (everyone involved in the process) will learn that things will run much smoother if they involve ops from the outset next time. And in most cases that is what they will do.

It sounds far-fetched, but it works, I have seen it. Please get in touch via the comments section and share your experiences.

End Notes

All definitions taken from AXELOS (2011) *ITIL Service Transition*. TSO, London.

About the author

Julia Harrison started from a technical background supporting and engineering end user technology solutions. About ten years ago she began to focus on service improvement and ITSM, and more recently Agile and Lean. Julia wrote this article during her time as Product Development Manager for ITSM at AXELOS Global Best Practice.

About AXELOS

AXELOS is a joint venture company co-owned by the UK Government's Cabinet Office and Capita plc.

It is responsible for developing, enhancing and promoting a number of best practice methodologies used globally by professionals working primarily in project, programme and portfolio management, IT service management and cyber resilience.

The methodologies, including ITIL®, PRINCE2®, MSP® and the new collection of cyber resilience best practice products, RESILIA™, are adopted in more than 150 countries to improve employees' skills, knowledge and competence in order to make both individuals and organizations work more effectively.

In addition to globally recognized qualifications, AXELOS equips professionals with a wide range of content, templates and toolkits through the CPD aligned AXELOS Membership and our online community of practitioners and experts.

Visit www.AXELOS.com for the latest news about how AXELOS is 'Making organizations more effective' and registration details to join AXELOS' online community. If you have specific queries, requests or would like to be added to the AXELOS mailing list please contact Ask@AXELOS.com.

Trade marks and statements

Sourced and published on www.AXELOS.com