SysAid

SELF-SERVICE TO THE RESCUE

by Stuart Rance and Stephen Mann

Many IT organizations see employee self-service as a ‘knight in shining armor’, ready to solve all their service desk issues in one quick project. The sad reality is that instead of the white knight they are hoping for, their investment in self-service often results in a white elephant, a possession that is useless or troublesome and which soaks up money and other resources without delivering much return on the investment.

In this paper we look at some of the reasons for thinking about self-service, what benefits it can bring to you and your customers, and what common problems you should plan to avoid. Hopefully this will help you to create real value with self-service, both for your customers and for your IT organization.

If you haven’t already seen it, then you may also want to watch this SysAid webinar on the same topic.

Customer and User Benefits of Employee Self-Service

Like every IT project, you should always start by considering the benefits for your customers[1] and users[2]. From this point of view, a great self-service implementation can deliver many benefits:

- **Faster access to help for users.** A well designed form, that is easy to fill in, can be much quicker and easier for a user than holding on the phone waiting to talk to an agent.
- **Improved user communication.** Users can check the status of incidents and service requests to discover the current status, how much progress has been made, and when they can expect things to be completed. Users can also enter information using their own words; what they say will be accurately captured and maintained as a permanent record.
- **Increased service hours.** Self-service can be available to users 24 hours a day, 365 days a year, even if you only staff your service desk during office hours.
- **Support for more languages and time zones.** Self-service can be translated into multiple languages much more easily than providing telephone support in every language. This can enable people to get first line support in their native language, with telephone assistance in a smaller number of languages.

[1] The term ‘customers’ is used in this paper to refer to the people in the business who fund IT services, and agree what service levels will be provided. focusing on the user and customer benefits listed above.

[2] The term ‘users’ is used in this paper to refer to the people in the business who make regular use of the IT services to help them do their work.

- **Fast incident resolution.** An incident that is resolved using self-service is typically completed much faster than one that is handled by telephone support.
- **Faster request fulfilment.** If service requests are fulfilled automatically, from initial request all the way to delivery and closure, then users can receive services much more quickly than if a manual process is used.
- **Better user experience.** Self-service can deliver a reliable, repeatable, and predictable level of performance. This allows an IT organization to be consistent in meeting user expectations for responding to and resolving incidents and service requests.

IT Benefits of Employee Self-Service

Self-service can also deliver significant benefits to the IT organization. Typically, benefits to the IT organization include:

- **Improved efficiency / reduced cost.** Incidents and service requests that are resolved with self-service require significantly fewer IT resources. This can save not only head-count but also equipment, office space, software licenses, etc.
- **Ability to leverage automation.** Self-service for service requests delivers most value when fulfilment of the request is automated. Ideally the entire value chain can be automated so that the user request is fulfilled with no manual intervention at all. Even if financial or other approval is needed to fulfil a request, this can be largely automated, with the request being approved by a business manager without needing IT involvement.
- **Better ability to handle high volumes of incidents when problems occur.** If a problem affects a large number of users, then this can result in lots of people trying to contact the service desk. A good self-service implementation can immediately direct these users to a simple one-click “me too” button, so that they can log their incident as one of many. Without self-service, the service desk could be inundated with calls, making response times very slow as well as adversely affecting the ability to handle incidents and requests not related to the current problem.

These benefits may be important to you, but you must start by focusing on the user and customer benefits listed above.

Is Self-Service Right for You?

The benefits listed in this paper can certainly be achieved, and they may be exactly what your organization needs, but you need to consider self-service along with all the other opportunities for investment, and make an informed decision about your investment priorities. Just because most organizations can benefit from self-service, doesn't necessarily mean that it's right for you, and it certainly doesn't mean that you should invest in it immediately! For example, if you are supporting expensive staff who generate significant revenue and have little expertise or interest in solving their own IT issues, you may want to continue to provide personal support to them indefinitely.

What's Included in Self-Service?

Self-service can include lots of different things. A typical self-service implementation includes many of the following:

- **The ability for users to log incidents.** Almost every self-service implementation includes front-end automation that allows users to log incidents for themselves. This may be via a web form, or using an app on their phone or tablet.
- **Provision of solutions to user incidents.** For example, the user could be offered an article that describes the issue they have, and explains how they can resolve it for themselves. The self-service portal should capture information about which solutions have been used, and whether the user was satisfied with the outcome, to enable reporting and improvement of these solutions.
- **The ability for users to log service requests.** This is almost always included in self-service. Ideally there's a service request catalog that is agreed upon with customers, and published to users. Self-service allows users to request things from this catalog. Typical requests might include:
    - Requesting access to a system, a service, or some data
    - Requesting software to be installed on a PC, phone, or tablet
    - Requesting new or replacement client hardware
- **Password reset capability.** There have been endless discussions about whether a password reset is an incident or a service request. We certainly don't intend to add to those discussions here, but regardless of how you categorize this activity it's important to include the ability to reset user passwords in your self-service capabilities. In many organizations, password resets can account for a very large percentage

of calls to the service desk and automating just this one thing can lead to a significant increase in productivity for both users and the service desk. An initial implementation of self-service might include just this capability, with more functionality being added over time.

- **Automated fulfilment of service requests.** Self-service delivers the maximum value when the entire end-to-end value chain is automated. For example, a user might request a software package and this could be automatically delivered to their PC or tablet; the process could include updating license information, charging the users business unit, and creation of audit trails to support software asset management. If management approval is needed for financial purposes, then this could also be automated, with managers using the self-service portal to approve (or disapprove) requests from their staff.

- **Access to FAQs and IT knowledge articles.** This should be in a range of formats, not just written documents. For example, there could be a video showing how to configure Wi-Fi on a smartphone. These FAQs and articles should help to answer the most common queries that the service desk has to deal with, and should be in a style and format that the users find helpful.

- **Access to business information and knowledge articles.** This could include guidance on how to complete a business process, such as submitting an expense claim or processing a common business transaction. It might include maps and travel directions for company locations or any other useful information. This could also include links to external articles and sites that are relevant to the users.

- **Status updates on outstanding incidents and service requests.** Self-service should enable users to check on the status of their incidents and requests. For this to be effective the underlying data needs to be present, which means that IT needs to regularly update records to show what progress has been made. Updates can be automatic or manual depending on the specific workflow.

- **Service status updates, and broadcast alerts.** When users connect to self-service they should be presented with information about the status of IT services, so that they can rapidly see if their current issue is related to an existing problem. The self-service portal can also be used to broadcast important information to users. For example, the portal can be one channel for communicating planned downtime for a service to users (but it may be important to use other channels to supplement this). It may also be useful to provide a service calendar, showing future releases, and any planned downtime. This could be integrated with a calendar that shows significant business events.

- **The ability to engage with service desk agents via other channels when needed.** For example, the self-service portal could include a "chat" option that allows the user to exchange text with a service desk agent. This can be a very efficient use of time as both the user and the agent can continue to carry out other tasks while the chat takes place.
- **Peer-to-peer support.** Many self-service implementations include the ability for users to offer and receive help from each other. For example, there may be discussion threads where users can comment on knowledge articles, or offer solutions to problems that don't yet have knowledge articles. This can extend to support for a range of communities, which may offer value far beyond IT self-service.
- **Management of personal IT asset information.** The self-service portal can provide the ability for users to check, and update, information about the IT assets they use, for example the model and serial number of their laptop or phone.

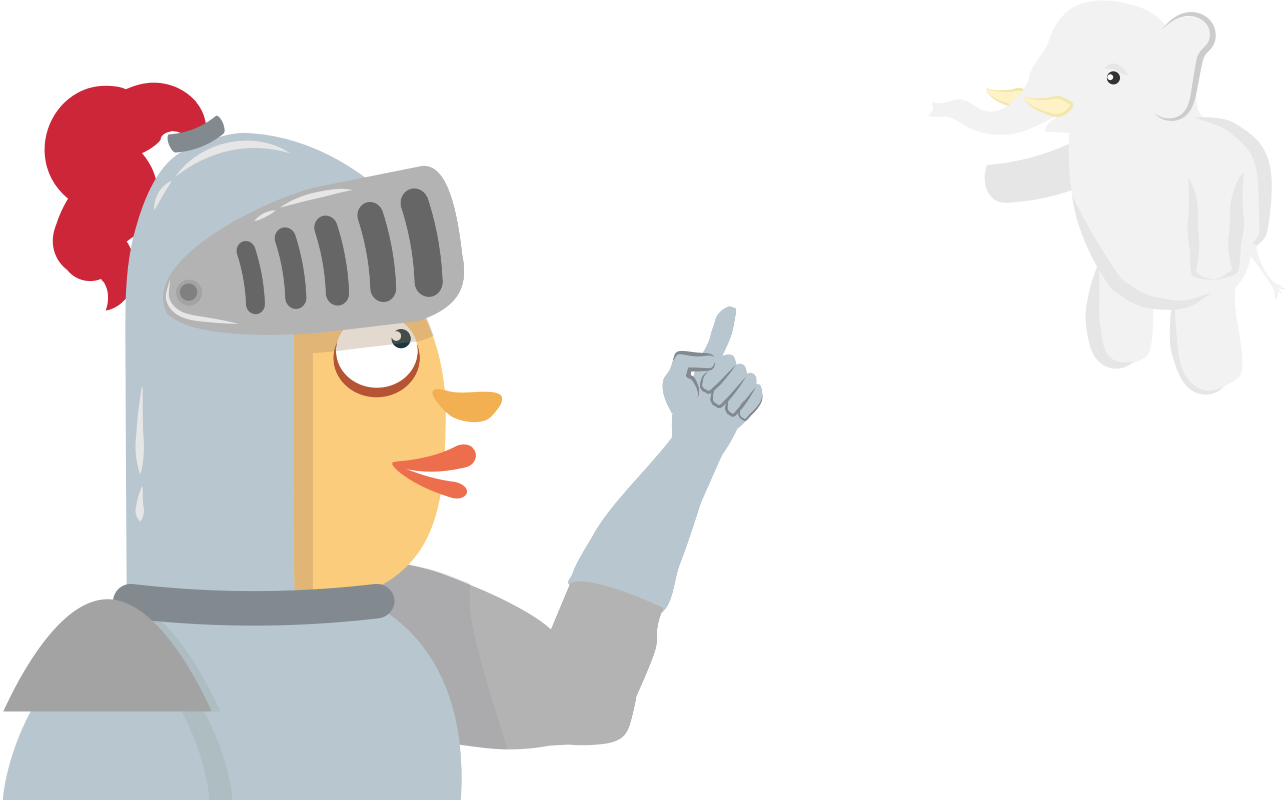
You don't have to offer all of this functionality in your self-service portal, and it would almost certainly be a mistake to try to include all of it in a first release as this would take a very long time to develop. You should consider all of these things and think about how much value they might have for your customers and users, and you may include many of them into your vision for self-service. You can then implement self-service in stages, but keep a clear vision of where you are trying to end up.

Remember that Self-Service Is a Capability, Not a Technology

With all of those benefits, you'd think that it would be easy to get a great ROI from an investment in self-service, but in fact many self-service projects result in increased costs, poor service levels, and unhappy customers. There are many different reasons for this, but the most fundamental mistake is treating self-service as a technology project. You certainly need to deploy some technology to support self-service, but this is not usually where difficulties arise, and if you spend most of your time and budget on the technology, then your project will almost certainly fail to deliver value. For your self-service project to succeed, what you do need to focus on is the other things that need to be in place to support the technology.

For example, a focus on customer experience is crucial. People will only make use of self-service if they have a good experience; they will stay away if they have a bad one. So every process, every screen, every touch point should be designed to create the best possible user experience. That way you will have willing volunteers, rather than reluctant conscripts.

Similarly, you need to ensure that the solutions you put in place are capable of modification to support potential future expansion of self-service. Once you have a portal in place, for example, other business units in your organization may wish to offer their services via the portal. Your design should be sufficiently flexible to ensure that you can add features and services that might be needed in the future.



Make Sure You Involve the Right Stakeholders

There are many stakeholders who should be involved in your self-service project. Some of these may only need to be involved at the start, or at other specific points in the project, but others should be included as part of the team developing the solution. Remember that self-service isn't primarily going to be used by IT people, so it should probably not be left to IT people to define and design the solution. Some of the key stakeholders that you should involve are.

- **Users.** It's essential to consult with users throughout the project. Ideally you should have some user representatives as part of the design team. These users will help to ensure that the self-service solution you deliver is going to meet their needs. It's the users, not IT, who should define and approve the look and feel of the user interface. Getting the right involvement from your users goes a long way in helping to ensure that self-service actually gets used.

- **Customers.** Your customers are responsible for defining and agreeing the IT services. They understand the relative priorities of different business processes, and they can make sure that you prioritize the needs of the correct users. If your service level agreement (SLA) includes metrics relating to response times and resolution times for incidents and service requests then you'll need to renegotiate these with the customers.

- **Financial controllers.** If you plan to automate resolution of service requests, then your organization's financial controllers will help you to ensure the approval process is acceptable to the business. If you just automate the technical aspects and leave financial approval as a manual process, then self-service will have many significant delays; it's much better to automate the entire end-to-end workflow.

- **Service desk, technical support, and application support.** The people who currently resolve incidents and fulfil service requests will help you to prioritize what should be included in early releases, and what can be left until later. They'll also have the knowledge and understanding of how things are done now, which is an important input to deciding how they should be done in the new world of self-service.

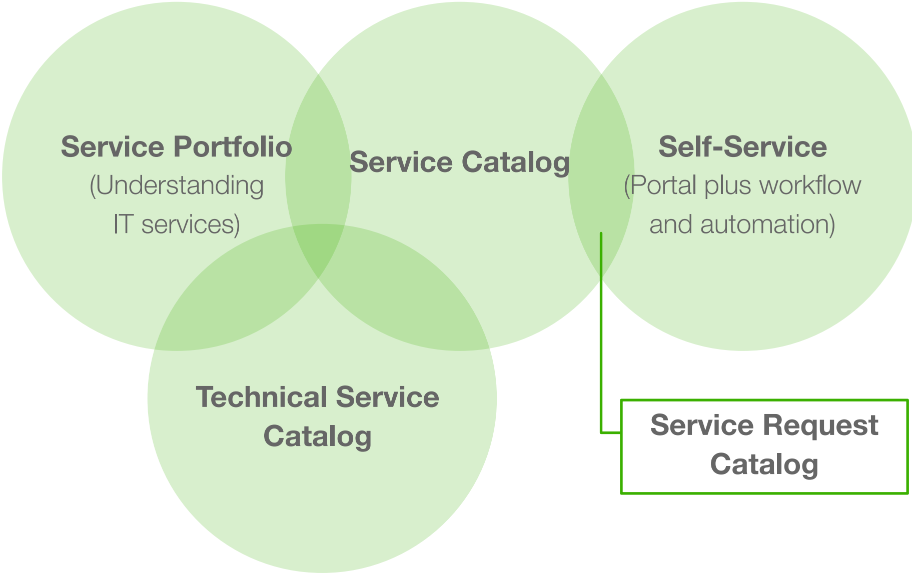
- **IT management and ITSM process owners.** Self-service could have an enormous impact on the way IT is managed within your organization. At a minimum, it's going to affect the processes you use for managing incidents and service requests, but it could potentially impact every other IT service management process. You need to make sure that the people who own these processes, and the people who are affected by them, are involved in the project.

Don’t Get Distracted by the Need for a Service Catalog

A *service request catalog* and a *service catalog* are not the same thing. If your organization doesn’t have a service catalog – listing everything you do in business terms – then you may be distracted by the similarity in name and think that creating a service catalog must form part of your self-service project. This can be a huge distraction, leading to a lot of additional effort with little direct impact on your self-service project.

It’s certainly very important to have a proper service catalog, so your customers can understand the business services that IT can deliver. But that can, and (generally speaking) should, be a separate effort from setting up self-service.

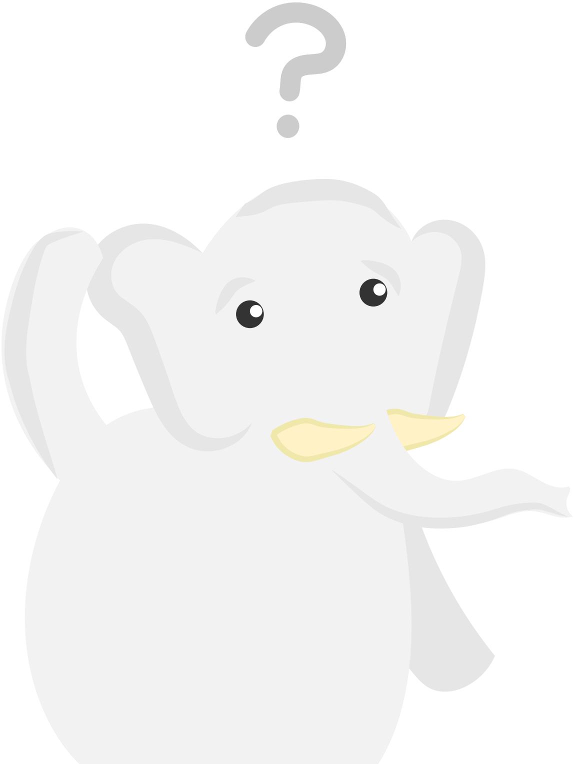
The diagram below shows the relationship between a service portfolio, a service catalog, a service request catalog, and a technical service catalog:


Service Portfolio
(Understanding IT services)
Service Catalog
Self-Service
(Portal plus workflow and automation)
Technical Service Catalog
Service Request Catalog

- **Service portfolio.** A view of all services, including services that are currently available, services that might be delivered in the future if you decide to invest in them, and retired services that you no longer supply to customers. The service portfolio considers services in terms of their cost and value, and is used to plan and prioritize investments.

- **Service catalog.** A customer-facing view of services that are available to the business. The service catalog can be used to market IT services, and to help customers understand what services they may wish to provide for their users. An example of an entry in the service catalog might be "user enablement" and this could include: provision of standard computer, tablet, and phone; with software and software licenses, support services, email, and file services delivered from a data-center; and user accounts to enable services to be used.
- **Service request catalog.** A user-facing view of service requests they can place. It's an important part of the self-service capability, which enables users to select the specific item they are trying to order. There is usually a one-to-many relationship between entries in a service catalog and entries in a service request catalog. For example, the "user enablement" service might allow the user to order specific models of a phone or PC, or to request the download of specific software packages.
- **Technical service catalog.** An IT facing view of capabilities that IT uses to deliver and support the services in the service catalog. For example, a "user enablement" service might be supported by Local and Wide Area Network services, Service Desk service, Backup service, etc.

You need to understand what service requests are available for your users as part of a self-service project. If you don't already have some sort of service request catalog then you may need to create something as part of this project. But it should not require a disproportionate allocation of time and other resources.



Avoiding Common Mistakes

We mentioned earlier that when self-service projects have disappointing outcomes it tends to be because they have focused on technology rather than capability. Here are some of the things that you need to think about to avoid the mistakes that lead to disappointment.

- **Involve customers and users in the design.** Customers know what value they want from self-service, which business processes should be prioritized, how much they want to satisfy user desires compared to how much they want to cut costs, etc. Users are the only people who know how they use self-service, what they use it for, how much time they are prepared to spend filling out forms, how many categories it's reasonable for them to choose between, when they'll run out of patience etc. Customers and users should not just be consulted for requirements; they should be actively engaged as members of the design team, providing input at every stage.

- **Define the scope or purpose of the project very carefully.** You need to be very careful when you define the purpose and scope for self-service. If you try to do too much in a single project then it takes far too long to deliver any value, and by the time it can be used it's unlikely to be what was needed. If the scope is too limited then users will rapidly learn that it does very little to help them, and they won't bother to make use of it. Once you lose their support any future expansion of the scope is likely to be ignored too. You need to strike a happy medium; just enough functionality to be really useful, but delivered fast enough that it's still relevant when it's delivered.

- **Make as much use of automation as you can.** If users can log incidents and service requests using a form, and then these are simply queued for eventual attention by someone on the service desk, self-service becomes just a new front-end for the service desk, delivering very few benefits. It's really important to automate as much as possible so that users experience rapid service. Otherwise self-service is seen as a place where low priority incidents and requests languish with no action, and it will fall into disuse.

- **Focus on user experience rather than cost saving.** If the only purpose of the self-service project is to save money by reducing the number of people answering phone calls, then it's very likely to fail even to do this because nobody will want to use it. If however the project focuses on improved user experience, it's very likely to deliver cost savings, as well as a better experience for users.

- **Make sure all stakeholders, including the users, understand "What's in it for me".** An important aspect of change management is to understand "What's in it for me" for all stakeholders. If IT simply create a self-service portal that they would like themselves, without identifying the value for other stakeholders, then the project will never succeed.

- **Provide enough high quality knowledge articles and make sure they are accessible and comprehensible to the people they are aimed at.** Self-service depends on the presence of really well documented knowledge articles that help users to resolve their own issues. These can be provided in a variety of formats, not just written documents but images, and audio and video recordings. They must be made available to users, be easy to search and locate, be available in the correct language(s), and be maintained so that they are up-to-date.

- **Provide encouragement for users to adopt self-service.** It's important to promote the value of self-service and to encourage users to try it. It's not enough to communicate with users just once, there needs to be an ongoing program to encourage users to make use of self-service. You have to do more than just send emails, you may need to use multiple communication channels, and you may need to provide training or other help for some users. Without a proper communication plan, most users will simply ignore the self-service option.

- **Give users a choice.** You may think that self-service is the ideal solution for all user incidents and requests, but there will be circumstances where users prefer alternative channels. It can be very frustrating for users when they have a simple query that they can't ask because it hasn't been anticipated by the people who designed the self-service solution. If you force people to use self-service with no choice, then you'll cause significant frustration and end up with dissatisfied users. A user may be stuck in a location with telephone access but no ability to use a self-service front-end, or may have a disability that prevents them from using self-service. They may simply have a strong preference for using another channel. If you provide your users with a choice, then you can encourage them to use self-service by making it work better for them than the traditional channels. This way they'll move to self-service by choice, and you'll get all the expected benefits without reducing customer satisfaction.

How to Run a Self-Service Project

The first thing to think about is what project management approach you want to take. This depends as much on the governance and culture of your organization as it does on the outcomes you want to achieve. Think about your vision and the steps you need to get there. It's usually a bad idea to try to deliver everything you might possibly want from self-service in a single monolithic project. Make sure you have an agreed vision and then think about the steps it will take to get there.

My preferred approach is usually to use an agile project methodology. Start by creating the minimum viable self-service solution as your first step. This needs to have enough value that your users will want to use it, but be sufficiently small that it can be achieved in a short timescale. Maybe you could start by implementing password reset capability, or maybe you could start by automating common service requests without putting any self-service in at all, still using your existing front-end. There are many different ways you could start, so you need to base this decision on your priorities, which you establish with your stakeholders.

Alternatively, you could run the self-service project using a big-bang (or waterfall) methodology. This does tend to take a lot longer before it delivers any value, but in some organizations it's the preferred approach. If you are going to use this approach then it's very important to get a thorough understanding of requirements very early in the project, and to avoid "scope creep" as the project progresses. Deliver the agreed functionality in the agreed time, and then consider requests for enhancements.

Some of the things you need to do include:

- **Find out what has worked for other IT organizations.** Discuss your ideas for self-service with other similar organizations. You may be able to do this by attending IT service management conferences, or by participating in discussions on social media, or by talking to your peers in other companies that you know.

- **Define clear objectives for self-service.** Self-service should be much more than just enabling users to log their own incidents, and providing them with a shopping cart. Think about what it means in business terms. You may have objectives about improvement in user experience, reduction in time to solve incidents, increased financial control for service requests, or even cost reduction. Make these explicit and decide how you're going to measure them.

- **Review existing support channels.** Make sure you understand how your users are getting help now. This should include reviewing data from the service desk about types and numbers of calls, but it should also include engaging with users to understand what they actually do. You may find that they are contacting support teams directly, or using informal peer-to-peer support, or just relying on their favorite internet search engine. You need to capture as much information as you can about how your users currently behave as a baseline, and to help you plan what is required.

- **Review existing service desk metrics.** What is being measured? What are the typical values? What are the trends? How do these compare to the SLAs? What do the customers and users think of them? After answering these questions, think about how these might be affected by the new self-service channel. For example, many IT organizations measure "First time fix rate", which is the percentage of calls to the service desk that are resolved during the initial call. Implementing self-service may cause this figure to become much worse, as the simple calls will no longer be going to the service desk. Similarly, figures for average time taken to resolve incidents may need to have separate targets for self-service and other channels, to recognize that each channel will have a different balance of calls.

- **Plan how to manage the knowledge that is needed for successful self-service.** Your self-service portal needs access to a lot of knowledge if it's going to provide value to your users. This means that your project needs to ensure that this knowledge is created, and that it will continue to be maintained and updated into the future. Start by talking to people on the service desk and in your support teams to find out what knowledge and information they currently use, what would make their lives easier, and what they think might be needed by users; also talk to a number of users about what they would find useful. Then identify who can provide the information you need. Think about what incentives you might need to encourage people to share their knowledge, and who has the ability to create knowledge in the format(s) needed. Remember, this should not just be writing articles, but could include creation of audio and video content as well. Finally, make sure that the project puts something in place to ensure that knowledge is continually refreshed and updated as part of your business as usual activity.

- **Design and implement the technology.** There certainly are many things other than technology that you need but it should go without saying that you do also need great technology to support your self-service project. The front-end user interface is the thing that your users and customers are most aware of, but this needs to be supported by sufficient automation that it actually delivers value. Start by automating things that you do frequently, and that you really understand. You should first simplify the activities, and then automate them, as this results in greater efficiency than simply automating your current manual process. Think about how the automation is going to interact with IT and business staff, for example if approvals are needed for a purchase, and ensure you optimize the end-to-end process, not just the self-service aspects.

- **Create a plan for management of organizational change (MoC).** The most difficult part of implementing self-service is getting people to change their behavior. This can only be done by focusing on MoC. If your organization has a preferred approach for MoC then you should use it; alternatively you can learn about John Kotter's 8-Step Process for Leading Change. Whatever approach you choose, it requires careful planning and plenty of time. Don't make the mistake of allocating a two-week timeslot in the middle of the project labelled "change organizational culture", as that is never going to work. One interesting approach that has worked well for a number of IT organizations is gamification. This means making a game out of people doing the things you want to encourage: awarding points for sharing knowledge or helping themselves, identifying leaders and people who are making good contributions, and encouraging the behaviors you want to see more of.

- **Keep the momentum going.** Implementation of self-service shouldn't just be a single project that results in a fixed set of deliverables. Keep talking to customers and users, and encourage them to offer you feedback, monitor how self-service is being used, capture detailed data about which parts are popular and which aren't, understand which users continue to use other channels and why, find out about planned changes to IT services and business processes, think about how you can expand self-service to provide more value, and plan improvements that will ensure your self-service continues to deliver value to your customers and users.

Summary

Self-service can create increased value, and reduce costs, for IT as well as for customers and users, but there are many things that can go wrong with a self-service project. The most important thing to understand is that self-service is not primarily a technology project, and the value it creates is much more than cost reduction. You need technology, and self-service will save you money, but the project should focus on customer experience and organizational change, rather than on technology and cost reduction; this way you'll have happier customers and users, as well as reduced costs.