It Takes a Village to Build an Image

R. Mark Koan
Arizona State University West
4701 W. Thunderbird Road
Phoenix, AZ 85069
(602) 543-8283
Mark.Koan@asu.edu

Kelly Caye
Arizona State University West
4701 W. Thunderbird Road
Phoenix, AZ 85069
(602) 543-8278
Kelly.Caye@asu.edu

Steven K. Brawn
Arizona State University West
4701 W. Thunderbird Road
Phoenix, AZ 85069
(602) 543-8295
Steven.Brawn@asu.edu

ABSTRACT
Managing operating systems and campus-supported software can be a daunting challenge. Many colleges and universities handle these challenges by using disk imaging software to create and deploy system images to some or all of their computers. ASU West has been building and deploying system images with Ghost for many years. Recently, however, we decided to re-evaluate our imaging processes and we found a number of ways to improve them. This presentation will share what we learned. It will discuss our procedures so that you can find ways to implement or improve your own strategies—whether you are starting from scratch or looking for ways to refine your current processes.

The IT department at ASU West uses Symantec Ghost to build images for virtually every Windows computer on campus. Our procedure is robust enough to ensure a well-tested image gets deployed consistently and flexible enough to accommodate images for a host of different environments. We not only deploy images in all of the IT managed labs and computer classrooms on campus. We also create and maintain a wide variety of specialty images for mobile computer carts, departmental laboratories, and check-out equipment. In addition, we also build and install images for the computers in faculty and staff offices.

Our presentation will outline our entire process for image creation and deployment from start to finish. We will discuss things like timelines, testing procedures, handling hardware differences, pitfalls, updating images, getting customer feedback, and deploying images in a variety of ways. Though we will focus primarily on the Windows environment, we can also provide some insights for Macintosh computers. The presentation will include documentation about our processes and some guidelines for building a successful strategy at virtually any other institution interested in this approach.

Categories and Subject Descriptors
K.6.3 [**Management Of Computing And Information Systems**]: Software Management – *software maintenance, software process.*

K.6.4 [**Management Of Computing And Information Systems**]: System Management – *Centralization/decentralization, Quality assurance.*
D.2.9 [**Software Engineering**]: Management – *Productivity, Software Configuration Management.*

General Terms
Management, Performance, Reliability, Standardization.

Keywords
Software Management, Imaging, Image Deployment, Software Deployment, Symantec Ghost, Ghost, Ghosting.

SIGUCCS'03, September 21–24, 2003, San Antonio, Texas, USA.

1. INTRODUCTION
The Information Technology department at Arizona State University West supports a wide variety of software. As at many universities, imaging software, specifically Symantec Ghost, is used to deploy most of this site-licensed, university-supported software.

Over the years, this process has been modified and refined in a number of ways. Our department took a close look at the entire process and evaluated the ways in which our use of imaging software could be improved. Through a systematic approach to image creation, testing, and deployment, we now have a much more robust and reliable software environment. This paper will outline ways in which these changes in the process have improved our software images and created a more stable and consistent software environment for our customers. Since this is one of the most visible aspects of the work we do in the IT department at ASU West and it is critical to the successful delivery of quality instruction, exploring ways to improve this process is time well spent.

2. IT'S ALL ABOUT IMAGE(S)
In 1997, using Ghost to create an exact disk image of the software installed on a computer's hard drive was already standard practice at ASU West. Ghost enabled the members of our Information Technology department to install an operating system and all of our campus-supported software on a single computer system and then create a perfect disk image of that computer. That image could then be deployed to any computer with the same hardware configuration and the computer could be up and running in a matter of minutes. Consequently, Ghost has been an invaluable tool on our campus for many years. On our campus, system

images are used on virtually every workstation including those in faculty and staff offices, in our student computing site, in our computer classrooms and labs, and for the presenter stations in our 'smart classrooms'.

In the early days, software images had to be created for each of these computers separately. A number of changes have allowed us to streamline the number of images we use, but the use of system images to control, maintain, and manage our software environment is more important than ever. There are a number of goals for our imaging process, but four of the most important are listed below.

1. Deploy software rapidly and consistently
2. Ensure that correct versions of required software are present on appropriate computers and functioning properly
3. Standardize the computing environment and security measures on campus systems
4. Recover computers from failure quickly

As software packages and the operating systems they run on grow in size and complexity, install times for each item on a system increase considerably. It is time-consuming to install all of the required software on one system—especially when one takes into account the amount of time required to ensure that the proper version of each software is installed and that all of the applications work together without conflict. It is simply not feasible to do that on every single computer for an institution with roughly 1500 computers.

The importance of correct versions and standardized software is even more pronounced in our computer classrooms. A wide array of different instructors must be able to depend upon having the proper version of the software to teach their classes and that it will perform as expected for every student. Training faculty and staff to use computers with a standard look and feel and a consistent set of applications is also easier. Standard software images also decrease the amount of time it takes to train new IT staff members and student workers to support our customers.

A standard software image, or images, also allows for globally defined security policies to be implemented across the campus on every system. Recent versions of Windows also allow for more robust security and customization through the use of profiles, but they contribute to a wide array of possible software conflicts with the operating system and each other. Testing software is incredibly important, but very time-consuming—particularly in a multi-user environment.

Finally, a thoroughly tested software image allows the members of our IT department to discover problems with various software configurations and find solutions before images are deployed to our customers. Once a stable image is built and configured properly, it can then be used to quickly prepare newly-purchased computers for deployment (or older computers for re-deployment), repair computers that have serious software problems, and recover systems that suffer critical hardware failure. Being able to test all of this in advance, establish software versions, and resolve possible issues means that images can be deployed much faster and much more consistently than manually installing software on each computer would allow.

For these reasons and others, Information Technology continues to use this software – now owned by Symantec Corporation – and this approach to managing our software environment. Over the years, we have learned a number of lessons about how to make our disk imaging efforts more efficient. This document outlines some of the significant changes we have implemented over the years and discusses ways in which they have improved our imaging process – and ultimately our level of customer service.

3. EVALUATING THE TOOLS

After many years of using Ghost, a team was formed in 2001 to identify the best means of imaging our supported Windows computers in faculty and staff offices, with an eye to a follow-up project evaluating the process for our public site computers. In the intervening years we had changed hardware vendors and new technologies had emerged that made it a reasonable time to re-evaluate our approach.

Dell had become our recommended hardware vendor and they had a program to deploy system software on computers before shipping them to us. We explored this option, but in the end, it did not turn out to be as flexible or timely as our environment demanded. Our impending campus-wide Windows 2000 implementation also offered a host of new technologies that looked promising including Rapid Install System (RIS) and Intellimirroring. A combination of new hardware recommendations and a new operating system environment provided the perfect opportunity to re-examine our approach and decide if it was still the most effective way to manage the software on our campus. And if so, could our current processes be improved? It was also important for us to consider whether or not it was feasible to use the same technology both in faculty and staff offices and in our public computing sites.

The team evaluated the following tools:

- Microsoft (http://www.microsoft.com) – RIS
- Symantec (http://www.symantec.com) – Ghost
- Altiris (http://www.altiris.com) – Deployment Server 5.1
- Storage Soft Solutions (http://www.storagesoftsolutions.com) – ImageCast
- PowerQuest (http://www.powerquest.com) – Drive Image Pro

These products were to be evaluated based on the following data:

- Comparison chart based on internally defined criteria
- Published reports
- Reviews
- Available feature lists

The team then defined a set of custom criteria that indicated our goals for the imaging process. These included criteria that addressed maximizing customer satisfaction, maximizing IT staff productivity, providing a means for customer self-install, and minimizing impact on the campus network infrastructure. Ultimately, the following issues were considered.

Criteria:

- Measure the time needed for deployment of image.
- Measure the down time involved for customer during configuration and setup phase.
- Determine number of IT staff members needed to image a computer(s).
- Determine the number of images that can be installed to different machines concurrently (multicasting).
- Determine what resources are available and what type of documentation is needed for end user to self-install image, e.g. media checkout, network boot disk, hard copy instructions, etc. Is self-install possible for our process?
- Determine and measure the network resources to be consumed by installation of image. i.e. network and server performance.
- Establish base for measuring customer satisfaction by surveying existing clients who recently have had new systems installed.
- Measure customers' expectations via surveys and interviews to determine what is expected, including the deployment down time, directions for installation, and ease of self-install process.
- Assess the value of other products that can handle the images with different hardware resulting in the reduction in the number of images needed.

This information was then put into a matrix and the products were evaluated based on the listed criteria.

As a result of this evaluation process, Altiris RapiDeploy and Symantec Ghost emerged as the clear frontrunners when measured against our criteria. When these results were considered in addition to other factors including staff expertise with the Ghost products and pre-existing licensing arrangements, it became clear that Ghost was the best solution for our needs.

4. THE IMAGING PROCESS

The team was then tasked with developing a process for creating, testing, deploying, updating, and maintaining software images for our environment. The remaining information is a description of the process developed and implemented by that team and a subsequent team that was appointed to look at the process for our public site images. Over the past two years there have been refinements to these initial processes and those suggestions and recommendations have also been included. It is important to remember that any imaging program evolves over time. You are unlikely to get it right the first time, but if you continue to look for ways to improve the process, you will continue to improve the efficiency of your department and the level of service you offer to your customers.

4.1 Establish Timelines

This is a step that often gets overlooked, but is probably one of the most common pitfalls in the imaging process. Clearly defined timelines are crucial for creating good images. This includes milestones The team that is building an image must know when they need to deploy the image and start counting back from there, so that they can establish firm dates for the other people who must participate in the process. On our campus, the consulting team has to know when their initial phase of software testing must be finished so that they can allocate enough time. The faculty members have to know when their requests must be submitted in order to have software installed in computer classrooms where they will be teaching. Last minute software installs are one of the most common causes of instability in system images. There also must be enough time for IT testing, and in the case of computer classroom images (more about these later), there must be time for faculty testing. Adequate time must also be allowed for researching problems that are discovered and finding solutions or workarounds.

Though it can be difficult to do, it is important to stand by drop-dead dates for images. If new software versions are released at the last minute, so be it. The software may be a welcome addition in the next image, but don't jeopardize the current one with an untested installation. If you include hastily-tested or untested software in the image that has already been tested, you are inviting trouble. Likewise, if you grant exceptions for one faculty member's last minute software request, you risk compromising the integrity of the image that all faculty members must use to conduct their classes. Being firm on this point is most difficult the first time around. Once you have established this policy and enforced it, faculty will likely adjust accordingly—after all, the bookstore has firm deadlines for ordering textbooks that they must respect; there is no reason why IT cannot have them as well. A sample of the schedule we use for the public site image at ASU West can be found in Table 1.

Table 1. Sample Public Site Imaging Schedule			
When (in business weeks, i.e., excluding the winter closure)?	#	**What?**	**Who?**
First Day of Pre-Reg or Registration: - late March for Fall - late October for Spring - early February for Summer	A1	Via an announcement, ask faculty & staff for any changes in SW requirements for the upcoming semester. The deadline for response is 5 business weeks before classes start. This would be in about the 3rd week of July for Fall, the 1st week of December for Spring and the 3rd week of April for Summer.	HL Supv
12th week before classes start: - last week in May for Fall - 2nd week in Oct for Spring - last week of Feb for Summer	B1	Determine what is needed for the upcoming semester, and report the decision to the IT Operations Manager: - minor changes to the current custom images; or - updates to the current baseline image; or - a new baseline image from scratch.	team
	B2	Create a semester-specific schedule, using this process overview as a template.	team
	B3	Through the Room Scheduling Office, schedule (reserve) classrooms for specific dates and times that will be needed in July, December, or April to build customized images. Also schedule specific dates and times that will be needed in August, January, or May to deploy images to the classrooms, and to perform "clean-up" on the computers that didn't take the image during normal deployment. The clean-up should take place the day after Ghost deployment. This will normally be during intersession, and for the computer classrooms, blocks of times will normally allow some computer classroom to be available for an event every day -- blocks of time will be staggered.	Ops-CLCC (for computer classrooms) Ops-Present (for non-CLCC classrooms)
11th and 10th weeks before classes start: - 1st & 2nd weeks in June for Fall - 3rd & 4th weeks in Oct for Spring - 1st & 2nd weeks of March for Summer	C1	Review and update as needed the documents that describe the steps to take to build baseline images and deploy final images. The baseline image documents are in *\\fsfab1\it\shared\operatns\Public Site Imaging\official docs,* and the deployment instructions are on this page.	team
	C2	Build new baseline images from scratch, if needed; OR Modify the existing baseline images if new images from scratch are not needed.	Consultant OR Ops-Builder
	C3	Notify IT staff of dates and location that the baseline images will be available for testing.	HL Supv
9th week before classes start : - 3rd week of June for Fall - 1st week of Nov for Spring - 3rd week of Mar for Summer	D1	Test the baseline images.	IT staff
8th & 7th weeks before classes start: - last week of June & 1st week of July for Fall - 2nd & 3rd weeks of Nov for Spring - 4th week of Mar & 1st week of Apr for Summer	E1	Identify changes needed in the baseline as a result of IT testing.	team
	E2	Create the final baseline image to incorporate identified needed changes, and notify team of its availability.	Consultant OR Ops-Builder

6th week before classes start: - 2nd week of July for Fall - 4th week of Nov for Spring - 2nd week of Apr for Summer	F1	Update the Classroom Software spreadsheet	HL Supv
	F2	Via an announcement, remind faculty and staff to identify any changed SW needs.	HL Supv
5th week before classes start: - 3rd week of July for Fall - 1st week of Dec for Spring - 3rd week of Apr for Summer	G1	Finalize the Classroom Software spreadsheet and the Public Site Software web page to reflect the software that will be installed in each public site.	HL Supv
	G2	Create presenter classroom, MCLAB, and delivery computer custom images.	Ops-Present
	G3	Create Technopolis and group room custom images.	Technop Supv
	G4	Create custom images for computer classrooms & Stat Lab.	Ops-CLCC
	G5	Notify IT staff and computer classroom instructors of dates and locations that custom images will be ready for testing.	HL Supv
4th week before classes start: - 4th week of July for Fall - 2nd week of Dec for Spring - 4th week of Apr for Summer	H1	Test custom images.	IT Staff & computer classroom instructors
	H2	Identify changes needed in the custom images as a result of IT and instructor testing.	team
3rd and 2nd weeks before classes start: - 1st and 2nd weeks of Aug for Fall - 3rd and 4th weeks of Dec for Spring - 1st week of May for Summer	I1	Create the final presenter classroom, MCLAB, and delivery computer custom images, incorporating changes resulting from testing	Ops-Present
	I2	Create the final Technopolis and group rooms custom images, incorporating changes resulting from testing.	Technop Supv
	I3	Create the final CLCC computer classroom and Stats Lab images, incorporating changes resulting from testing	Ops-CLCC
	I4	Deploy the presenter classroom images to all smart presentation classrooms except those in CLCC, to MCLab cart computers, and to delivery computers.	Ops-Present
	I5	Deploy the Technopolis classroom images to all Technopolis computers.	Technop Supv
	I6	Deploy the presenter classroom images to all student group meeting rooms.	Technop Supv
	I7	Deploy the computer classroom and Stats Lab images in CLCC.	Ops-CLCC
	I8	Deploy the presenter classroom images to presenter classrooms in CLCC.	Ops-CLCC
One week before classes start	J1	Ensure all images are ready in the classrooms. “Clean-up” Ghost images to ensure the new image is on all computers, except where hardware problems make it not possible. Empty the FSAdmin group in w.it.classrooms OU	Ops-CLCC, Ops-Present, Technop Supv HL Supv
During 1st week after the start of the new semester	K1	Clean up old files from the Ghost server - delete folders for all but the current (new) semester and the prior semester	Ops-CLCC

4.2 Image Creation

The first step is to create what we call a "baseline" image. This will be the foundation for all software images that are created; if this foundation is functional and stable, it will make troubleshooting the more complicated images much easier. A great deal of time and effort goes into this step, but breaking it into manageable phases and paying close attention to sequencing helps avoid countless problems that would otherwise arise later on in the process. It is important to note that this is done twice. At ASU West, two completely separate baseline images are created to handle the challenges of single user systems (in faculty and staff offices) and multi-user systems (in the public computing sites and computer classrooms). These differ in important ways and one image does not serve both needs well. Nonetheless, the process for creating each image is very similar. The differences have more to do with configuration, settings, security and the actual software that gets installed than with the process itself.

4.2.1 Define Software

Before an image can be created, the specifics of the software to be deployed must be agreed upon. At this point, the software selection should only include the most basic elements that will be required on every system. This is only a baseline from which other images will be created. A typical image would include the OS and all of the basic productivity applications supported in a particular environment. An example of some standard items can be found below.

- Operating system
- Anti-virus software
- Office suite
- Web browser(s) & plugins
- Email software
- FTP client
- Compression/decompression utility
- Multimedia player(s)

These are the applications that will reside on every computer in your environment and this will vary from school to school. Regardless, it should be your most fundamental software that is typically purchased under a site-license by your institution for all users. Specialized applications for specific classrooms, labs, or other needs will be added later. This phase of image creation requires considerable planning in order to standardize the operating system and any service packs that will be installed, the software applications that will go into the image, and a specific version number for each program, including any patches that will be applied.

4.2.2 Test Software

Once a list of supported software has been compiled, each application needs to be individually tested. At ASU West, the IT consulting team is responsible for the initial phase of testing and for recommending product version numbers, patches, and service packs. The consultants test each application and look for solutions to any problems that are uncovered. Then they make recommendations about which versions to support (sometimes newer versions are not recommended, depending on the results of testing). These decisions must be made prior to the initial creation of an image, so timing is critical. Once the decisions are made, an up-to-date list is put together for the image creation process.

4.2.3 Install Software

After all the software and versions have been agreed upon, it is time to start the installation. This is a fairly involved process and must be documented thoroughly. When building an image there will invariably be countless tweaks, patches, and fixes that are applied to make it behave properly. It is also wise to document any configuration settings or customization information about the programs being installed. Another important issue is sequencing: document the order in which software is installed so that results can be consistently reproduced. Once this step is completed, it is time to use a program from Microsoft called Sysprep (Sysprep will be discussed in greater detail in section 4.5.2). This tool is an invaluable utility for a robust imaging program; once it has been configured and used on the image, the image is ready to be tested.

4.3 Image Testing

Everyone in the IT department is asked to test the initial baseline image and report his or her findings. Some of the responsibilities are divided up along lines of expertise. The network services team is tasked with evaluating the networking configuration and the overall security settings. Consulting is responsible for evaluating each piece of software and verifying that it works as expected and with the other software that is installed. Each member of the consulting team is a designated expert on a number of pieces of software that we support and he or she is responsible for testing those applications in the image. The operations team can also participate in testing at this point, though their biggest responsibilities come later.

In the case of computer classroom images, we have an open testing process and encourage anyone using our software images in public sites to come and test the images. This allows the faculty members who will actually be teaching with these systems to test them and give feedback. This step is very important. It gives customers a say in the final product they will be using and it enables IT to find out about problems that might have otherwise been overlooked. It is always easier to correct these problems before an image has been deployed.

Once the image has been thoroughly tested, all results are sent to the imaging team and they research any problems that have been discovered and try to find solutions. Once they have addressed all of the problems, they create a new image that will be the final baseline for all other images.

4.4 Image Replication

At this point, the imaging process forks significantly. One of the most difficult challenges for any robust imaging program is the variety of hardware and software configurations that must be supported. Our experience has helped us uncover ways to manage multiple images effectively and streamline wherever possible. We use one baseline image for office computers and laptops and another for all of the public site computers. From these two starting points, we are able to create a wide array of images and avoid needless duplication of work.

4.4.1 Office Images

The baseline image created for faculty and staff office computers is ready to be deployed after it has been finalized. It is dropped on to every new computer as it comes in as part of the setup process for a new computer system. It also allows us to re-image older systems (assuming their older hardware meets all of the minimums for the software on the new image.) and update them very quickly. This re-imaging of older system can happen by request or in the event of catastrophic system failure. Sysprep allows the technician to enter the personal information required at setup and then the user is ready to go (more about this later). After adding the user as a local administrator on the system, joining it to our domain, creating an Outlook profile, and copying any backed up data and settings, the customer is ready to use their new system in fairly short order. All additional non-standard software is installed at the desktop, either via the self-install service IT offers or by a student technician.

4.4.2 Public Site Images

The computers that are deployed in our public sites are a different matter entirely. Images for these computers are created from a completely separate baseline. This baseline is developed in much the same way as the office baseline, but with a few important differences. The general look and feel of this image is somewhat different, as there are instructions on the desktop for logging in as a "guest" user in addition to links for convenient access to frequently needed information. There are also some important 'behind the scenes' changes, related to security, authentication, printing and so on. The baseline that is developed for this purpose is then used as a platform to build a wide variety of images.

This image is the foundation of different images that go into the computer classrooms (each with unique software depending on the courses taught in that particular classroom), Technopolis (our student computing site), presenter stations in our 'smart classrooms', and various departmental labs.

This means that a solid baseline is absolutely essential. Once that is established, however, there is still much work to be done. The operations team now begins testing in earnest. It is their job to certify two important aspects of the images. First, the image must function properly in a multi-user environment. This means that applications must have the appropriate security settings and permission levels, profiles must not conflict with one another, performance must not degrade as the number of profiles on the system increase, etc. Of course, the expertise of the consultants and network services staff is available during this phase, but members of the operations team are the experts in the multi-user environment, so the bulk of this testing falls to them. The second task they undertake is testing all of the specialty software that is used in one classroom or one lab but is not deployed in other areas of our campus. This software is deemed 'special use' software and is used only on a limited subset of campus computers. The team's job is to make sure that the special use software not only functions properly, but also that it does not break the existing software that is already installed on the baseline image.

4.4.3 Hardware Differences

Another challenge for the creation of software images that are to be deployed across multiple computers is the challenge of differing hardware. The magnitude of this challenge will vary from school to school depending on how tightly your hardware purchasing is controlled. At ASU West, we recommend Dells, and though we cannot and do not force any person or unit to purchase our recommended configurations, most of the campus follows our recommendations closely so that they can receive the highest level of support from IT. Obviously, there are exceptions, but for the most part we have a campus full of Dell hardware of varying ages. It is *much* easier to implement a robust imaging program with standardized hardware. While we were transitioning from our old vendor, Tangent, to our current vendor, Dell, we had to build, test and maintain a much larger array of system images. We are on a 3-4 year hardware cycle and so now most of the Tangents have been phased out, allowing us to streamline our process even further.

In our experience, there are some key factors when trying to build system images that will deploy across multiple hardware configurations.

1. Building an image on the latest hardware insures the greatest chance of backward system compatibility
    a. For Windows systems we use a current desktop computer
    b. For Macintosh systems we use the latest laptop computer
2. Building a minimum driver installation and including a folder containing the drivers for hardware variants enables hardware to be installed quickly after a system has been imaged.
3. Standardized hardware recommendations are vital
4. Sysprep is your friend.

One of the most important tools for decreasing the number of images you must create and maintain is a Microsoft utility called Sysprep. According to Microsoft, three things must be identical for this to work across multiple computers. They must (1) have the same Hardware Abstraction Layer (HAL), (2) have the same mass storage controller, and (3) support the Advanced Configuration and Power Interface (ACPI). Fortunately, we have discovered some ways around some of these limitations. We have been able to deploy an image built on a desktop computer across multiple desktop *and laptop* configurations. Obviously, these computers have different HALs, so do not immediately surrender if you don't think you can meet criterion number one. The second criterion is critically important, but fortunately, mass storage controllers do not all have to be the same. What *must* be the case is that all of the drivers for the mass storage controller (the IDE controller on the motherboard in most cases) have to be included in the Sysprep .inf file. It will support multiple sets of drivers, and this has been key for minimizing the amount of images we have to create and maintain. Unfortunately, we have not found a workaround for computers that do not support ACPI,so we were not able to use our Sysprep images successfully on our older Tangent computers, but even our earliest Dells support this option, though it must be enabled in the BIOS before imaging the system. With newer versions of Windows and Sysprep, Plug and Play devices, such as network adapters, modems, video adapters, and sound cards, do not have to be the same on the master and target computers. Using Sysprep when creating images decreases deployment time dramatically compared to standard or scripted installations.

4.5 Deploying Images

There are a number of ways to deploy software images and each will have different appeal depending on the environment. ASU West uses a variety of deployment methods. We currently deploy images using a GhostCast server, bootable CDs, and bootable DVDs. Which technique is best depends on a variety of factors specific to the particular systems being imaged.

Many of our public sites systems are imaged over our network using the GhostCast server and boot disks. This allows for an entire computer classroom or the entire student computing site to be imaged at one time. This is very convenient, but also places a significant load on the network and must be done with caution. There can be problems crossing switches (particularly from different vendors), and synchronizing settings among routers, switches, hubs, and network adapters is very important. This method also generally requires the use of a dedicated Ghost partition or a floppy disk that must be manually inserted into each computer.

Our office computers are primarily imaged from a set of bootable CDs. Since there are some steps that must be completed by an IT technician anyway, this works out very well and minimizes network traffic. It is a flexible, portable, fast solution and works very well. In the case of re-imaging, it also allows us to get a signed consent form explaining to the user that this process will wipe all data from the hard drive and anything that has not been backed up will be lost. Due to the size of current images (our latest requires 3 CDs), we are in the process of moving to bootable DVDs for this deployment method.

4.6 Image Updates

Of course, creating the image is only part of the battle. Once images are created, they have to be updated, maintained, and archived. We have a set schedule for the creation and deployment of images. Images are created three times a year. One image is created for the fall semester, one for the spring semester, and one for the summer sessions. This is true for the office images and the public site images. If new software becomes available in the interim, faculty and staff may install or request it individually for their office computers, but the office image is not updated until the next cycle, nor is the new software deployed in the public sites until the next image is created. Major changes are made to the office image at least one semester prior to implementing them in the public sites. This strategy allows faculty and staff to become acquainted with new software before they are asked to teach and assist students with it. There are two standing teams charged with the creation of the initial baseline images. One is tasked with creating the office image and another with the public site image. Though the entire department participates in the process, these teams are responsible for keeping track of tasks and deadlines, finding solutions to problems uncovered in the testing phase, and producing and updating the images.

5. COLLABORATION

Virtually everyone in our IT department has a hand in creating, testing, updating, and deploying images. Software images are critical to the way we conduct business, so their quality must be a top priority. We have found that producing a top-quality image requires the expertise of virtually everyone in the department. In the past, one or two people would be tasked with creating images for their area. Not only was a lot of work duplicated, but there was tremendous lack of standardization. Versions varied widely, passwords were not consistent, installation options were not standardized, and thorough testing was not done. Reliability was perhaps the biggest casualty of this approach. It is simply too big a job for one or two people. Getting the entire department involved has been one of the most important factors contributing to our success. It really does take a village (or at least an entire IT department) to build a good software image.

6. DOCUMENTATION

Another critical piece of any good imaging process is the undeniable need for good documentation. If your department is anything like ours, this is no small order. Nonetheless, it is absolutely crucial for a successful program. Items that are not documented are missed in future images or when staff changes. Document everything thoroughly and clearly—no matter how painful that is— and it will greatly improve your imaging process.

Document a set of guidelines for the creation of a software image. In our experience, steps that are omitted from the documentation will sooner or later be omitted from an image. Make sure documentation is thorough, clear, and complete. Someone should be able to pick up the documentation, read through it, and repeat the results consistently every time. It is also important to document what goes into each image so that you have a record of what software was installed on each image. We have also recently started an organized archiving system to keep old images available for a period of time. They are not often needed, but occasionally, they can come in very handy.

7. CONCLUSION

This process for creating, testing, and deploying software images has improved the reliability, stability, and consistency of our software environment. It permits us to safely distribute consistent images across the computer classrooms, the student computing sites, and faculty and staff offices—regardless of who is deploying the image or leading the image team that semester. Our customers are served more quickly, and they have a much greater likelihood of having the applications they need where and when they need them. They can also have a much higher degree of confidence that those applications will work as expected because of the thorough testing process that goes into each image build.

Like many institutions, ASU West has used disk imaging software for a long time to help manage our computer systems. It is an effective way to install and repair systems quickly. By making the process even more thorough and systematic over the last two years, we have reduced the number of help calls we receive, resolved those we do receive more rapidly, and can now offer a higher level of service to our customers than ever before. Even schools with long-standing imaging programs are likely to benefit from re-evaluating the way they build and deploy those images. Our images and our processes improve with each iteration. I guess it just goes to show that there is always room for improvement—even for long-standing procedures… perhaps *especially* for long-standing procedures.