

## 6 SYSTEM IMPLEMENTATION

### Purpose

The purpose of **System Implementation** can be summarized as follows: making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the Performing Organization (the transition). At a finer level of detail, deploying the system consists of executing all steps necessary to educate the Consumers on the use of the new system, placing the newly developed system into production, confirming that all data required at the start of operations is available and accurate, and validating that business functions that interact with the system are functioning properly. Transitioning the system support responsibilities involves changing from a system *development* to a system *support and maintenance* mode of operation, with ownership of the new system moving from the Project Team to the Performing Organization.

A key difference between System Implementation and all other phases of the lifecycle is that all project activities up to this point have been performed in safe, protected, and secure environments, where project issues that arise have little or no impact on day-to-day business operations. Once the system goes live, however, this is no longer the case. Any miscues at this point will almost certainly translate into direct operational and/or financial impacts on the Performing Organization. It is through the careful planning, execution, and management of System Implementation activities that the Project Team can minimize the likelihood of these occurrences, and determine appropriate contingency plans in the event of a problem.

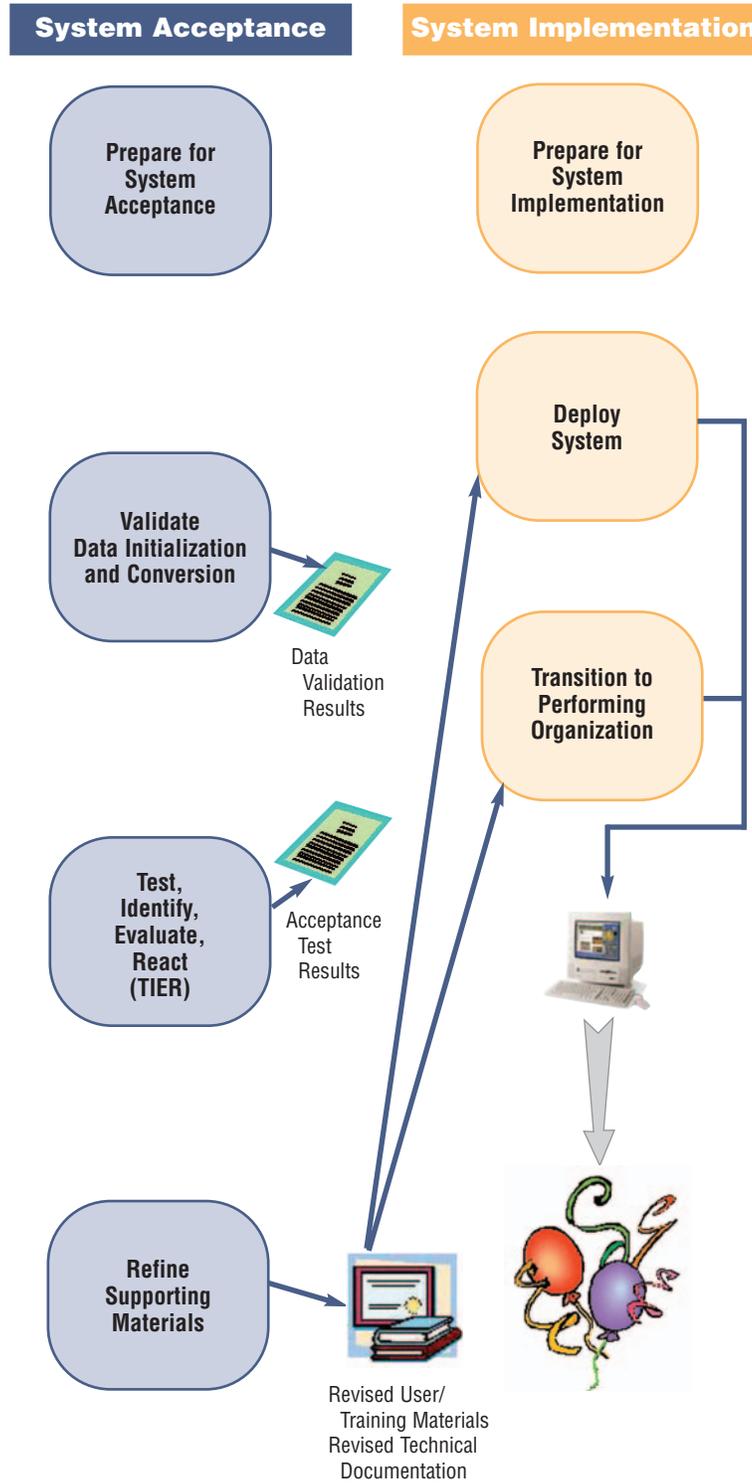
## List of Processes

This phase consists of the following processes:

- ◆ **Prepare for System Implementation**, where all steps needed in advance of actually deploying the application are performed, including preparation of both the production environment and the Consumer communities.
- ◆ **Deploy System**, where the full deployment plan, initially developed during System Design and evolved throughout subsequent lifecycle phases, is executed and validated.
- ◆ **Transition to Performing Organization**, where responsibility for and ownership of the application are transitioned from the Project Team to the unit in the Performing Organization that will provide system support and maintenance.

The following chart illustrates all of the processes and deliverables of this phase in the context of the system development lifecycle.

Figure 6-1



### List of Roles

The following roles are involved in carrying out the processes of this phase. Detailed descriptions of these roles can be found in the Introductions to Sections I and III.

- ◆ Project Manager
- ◆ Project Sponsor
- ◆ Business Analyst
- ◆ Data/Process Modeler
- ◆ Technical Lead/Architect
- ◆ Application Developers
- ◆ Software Quality Assurance (SQA) Lead
- ◆ Technical Services (HW/SW, LAN/WAN, TelCom)
- ◆ Information Security Officer (ISO)
- ◆ Technical Support (Help Desk, Documentation, Trainers)
- ◆ Customer Decision-Maker
- ◆ Customer Representative
- ◆ Consumer
- ◆ Performing Organization
- ◆ Stakeholders

### List of Deliverables

The following table lists all System Implementation processes, some techniques available for use in executing these processes, and process outcomes and deliverables.

Figure 6-2

<b>Processes</b>	<b>Techniques</b>	<b>Process Deliverables (Outcomes)</b>
Prepare for System Implementation	Interviews Distribution of Materials Coordination of Training Logistics	<i>Established Team and Environment for System Implementation</i>
Deploy System	Training Sessions Manual Business Operations Parallel Operations	<i>Migrated and Initialized Data Operational System</i>
Transition to Performing Organization	Training Sessions Phased Ownership	<i>Ownership of System by Performing Organization</i>

## 6.1 PREPARE FOR SYSTEM IMPLEMENTATION

### Purpose

The purpose of **Prepare for System Implementation** is to take all possible steps to ensure that the upcoming system deployment and transition occurs smoothly, efficiently, and flawlessly.

### Description

#### Roles

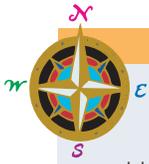
- Project Manager
- Project Sponsor
- Business Analyst
- Data/Process Modeler
- Technical Lead/Architect
- SQA Lead
- Technical Services
- Information Security Officer
- Technical Support
- Customer Decision-Maker
- Customer Representative
- Consumer
- Performing Organization
- Stakeholders

In the implementation of any new system, it is necessary to ensure that the Consumer community is best positioned to utilize the system once deployment efforts have been validated. Therefore, all necessary training activities must be scheduled and coordinated. As this training is often the first exposure to the system for many individuals, it should be conducted as professionally and competently as possible. A positive training experience is a great first step towards Customer acceptance of the system.

During System Implementation it is essential that everyone involved be absolutely synchronized with the deployment plan and with each other. Often the performance of deployment efforts impacts many of the Performing Organization's normal business operations. Examples of these impacts include:

- ◆ Consumers may experience a period of time in which the systems that they depend on to perform their jobs are temporarily unavailable to them. They may be asked to maintain detailed manual records or logs of business functions that they perform to be entered into the new system once it is operational.
- ◆ Technical Services personnel may be required to assume significant implementation responsibilities while at the same time having to continue current levels of service on other critical business systems.
- ◆ Technical Support personnel may experience unusually high volumes of support requests due to the possible disruption of day-to-day processing.

Because of these and other impacts, the communication of planned deployment activities to all parties involved in the project is critical. A smooth deployment requires strong leadership, planning, and communications. By this point in the project life-cycle, the team will have spent countless hours devising and refining the steps to be followed. During this preparation process the Project Manager must verify that all conditions that must be met prior to initiating deployment activities have been met, and that the final 'green light' is on for the team to proceed.



More than at any other point in the project, the Project Manager must plan for failure, and must have a defined set of contingency plans to be executed in the event of a problem encountered during deployment. Stakeholders and all key decision-makers must clearly understand and agree to the various "go/no go" criteria by which decisions will be made whether or not to proceed with the deployment. In the event of a failure, time lost as a result of an ill-defined course of action can be costly not only in terms of Project Budget, but equally important, in terms of Customer and Consumer confidence.

The final process within the System Development Lifecycle is to transition ownership of the system support responsibilities to the Performing Organization. In order for there to be an efficient and effective transition, the Project Manager should make sure that all involved parties are aware of the transition plan, the timing of the various transition activities, and their role in its execution.

Due to the number of project participants in this phase of the SDLC, many of the necessary conditions and activities may be beyond the direct control of the Project Manager. Consequently, all Project Team members with roles in the implementation efforts must understand the plan, acknowledge their responsibilities, recognize the extent to which other implementation efforts are dependent upon them, and confirm their commitment.

## 6.2 DEPLOY SYSTEM

### Purpose

The purpose of the **Deploy System** process is to perform all activities required to successfully install the new system and make it available to the Consumers.

### Description

Deploying the system is the culmination of all prior efforts – where all of the meetings, planning sessions, deliverable reviews, prototypes, development, and testing pay off in the delivery of the final system. It is also the point in the project that often requires the most coordination, due to the breadth and variety of activities that must be performed. Depending upon the complexity of the system being implemented, it may impact technical, operational, and cultural aspects of the organization. A representative sample of high-level activities might include the installation of new hardware, increased network capabilities, deployment and configuration of the new system software, a training and awareness campaign, activation of new job titles and responsibilities, and a completely new operational support structure aimed at providing Consumer-oriented assistance during the hours that the new system is available for use (to name a few). Whatever the realm of activities related to the new system, their impacts should be addressed in the Organizational Change Management Plan, while specific deployment activities should all be encompassed in the Project Implementation and Transition Plan, (both created during the Project Planning phase of the Project Management Lifecycle.)

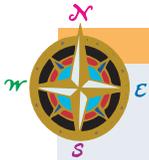
#### Roles

- Project Manager
- Project Sponsor
- Business Analyst
- Data/Process Modeler
- Technical Lead/Architect
- SQA Lead
- Technical Services
- Information Security Officer
- Technical Support
- Customer Decision-Maker
- Customer Representative
- Consumer
- Performing Organization
- Stakeholders



Depending upon the environment or the type of system being implemented, this phase may also warrant additional activities including 'sunsetting' (retiring) any related legacy systems, executing parallel runs, and managing external communications.

All Consumer training should be performed prior to physically migrating the system to the production environment. This will enable the Consumers to begin to familiarize themselves with the system, and will help to establish their expectations regarding what the system will and will not do.

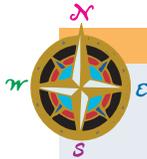


When it comes to training, sometimes the timing of the training can be as important as the content. Conducting the training after the system has been rolled out to the Consumers may cause them to form poor perceptions of the system, simply due to the difficulties associated with an unnecessarily lengthy learning curve. Similarly, holding the training sessions too far in advance of the deployment presents Consumers with the challenge of having to recall what was taught, again leading to possible frustration and unhappiness with the system.

The sequencing of deployment activities is just as important as it was with previous testing activities. This sequence of events should be encompassed in the Deployment and Transition Plan section of the Technical Specification, and will address and prioritize any necessary training activities, set-up activities needed to prepare the production environment (desktop, LAN, servers, data center, etc.), and data conversion and validation activities. This deployment plan will also define the steps for physically migrating the system and any associated utilities to production, and for validating the accuracy and completeness of this migration after these steps have been performed. During deployment, Project Team members may often be required to work extra hours to meet aggressive timeframes, or additional staff may be brought in temporarily to assist with large data entry efforts. Proper planning and sequencing of the deployment can help to minimize these situations, and reduce the chance of any missteps that could result in having to restart the deployment process, or lengthen the implementation schedule.

As the system is enabled, and the Project Team validates that the application is performing to expectations, there may be times when certain system functions seem suspect. One of the challenges most frequently faced by Project Teams is to determine the root cause of potential issues. Discrepancies that exist within the data could be due to defects in the application's business logic, or could be the result of data that was improperly migrated into the system. Similarly, the inability of a Consumer to access specific features of the system could be caused by improperly configured hardware, or incorrectly

established security privileges. To minimize confusion and reduce the opportunity for such issues to surface, every attempt should be made to immediately validate each step of the deployment as it is performed.



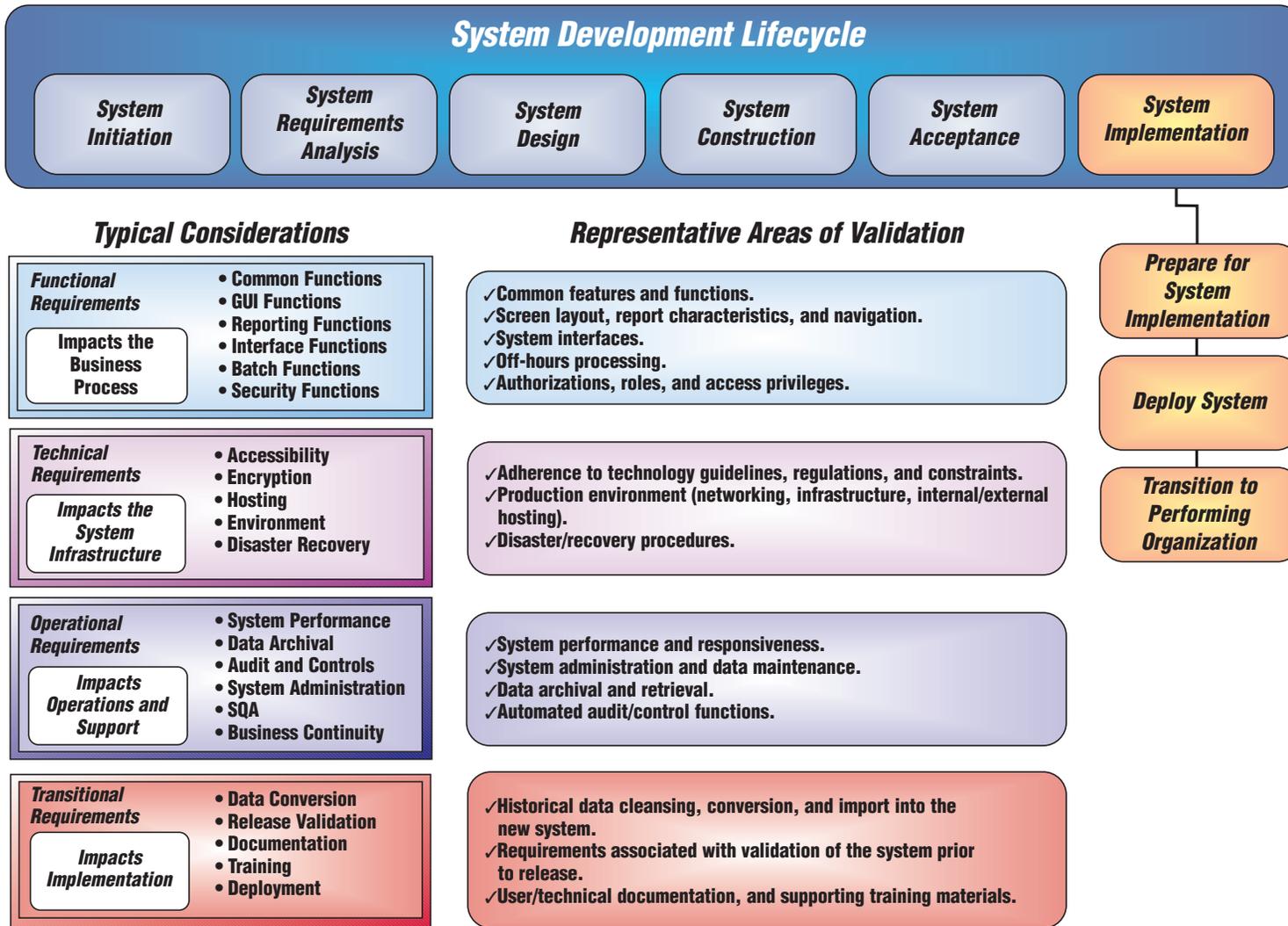
Ideally, there will be no aspect of the implementation that was not previously tested during System Acceptance. Whether or not this is true, there is always the possibility that routines or utilities that worked properly in one environment may not work identically in another. With this in mind, the Project Team should always validate the success of each step of the deployment, and wherever possible, should take appropriate steps to enable the team to “fall back” to a prior state should the severity of a problem warrant such an action.

Additionally, the Project Manager should be aware that not everyone is open or receptive to change. As a system is rolled out to its target audience, the team must remain keenly attentive to how it is perceived. The fact that functions that were present in the legacy system no longer exist or work differently may cause some Consumers to see the new system negatively. And while the new system may provide overall benefits to the business or agency, those benefits may come at the expense of additional work responsibilities to some of the individuals who interact with the system (e.g., the new system may require the entry of additional data that was not previously required). By understanding some of the dynamics behind how the system is being received, the Project Team may be better able to identify or publicize some of the benefits that the system provides. A well-defined Organizational Change Management Plan should have anticipated and addressed these issues.



One effective way to gauge the use and acceptance of the system is for the Project Team to maintain open communications channels with the Technical Support or Help Desk operation, if one exists. This will provide a broader view of potential issues or suggestions that can then be addressed proactively.

Figure 6-3 System Implementaion Considerations



### 6.3 TRANSITION TO PERFORMING ORGANIZATION

#### Purpose

The purpose of the **Transition to Performing Organization** process is to successfully prepare the Performing Organization to assume responsibility for maintaining and supporting the new application.

#### Description

In many organizations, the team of individuals responsible for the long-term support and maintenance of a system is different from the team initially responsible for designing and developing the application. Often, the two teams include a comparable set of technical skills. The responsibilities associated with supporting an operational system, however, are different from those associated with new development.

In order to effect this shift of responsibilities, the Project Team must provide those responsible for system support in the Performing Organization with a combination of technical documentation, training, and hands-on assistance to enable them to provide an acceptable level of operational support to the Consumers. This system transition is one element (albeit a major one) of the overall Project Implementation and Transition Plan, developed as part of the PM Lifecycle. The Project Manager should review the transition plan to confirm that all defined actions have been successfully completed.

#### Roles

- Project Manager
- Project Sponsor
- Business Analyst
- Data/Process Modeler
- Technical Lead/Architect
- SQA Lead
- Technical Services
- Information Security Officer
- Technical Support
- Customer Decision-Maker
- Customer Representative
- Consumer
- Performing Organization
- Stakeholders



One common approach to successfully transitioning support responsibilities is to implement a phased transition schedule that gradually shifts increasing ownership of the system to the Performing Organization. Early phases would have the Performing Organization's support and maintenance team primarily observing the Project Team as part of knowledge transfer, while later phases would have the support team acting as the first line of response. The exact phases and their timing should be determined as part of transition planning earlier in the lifecycle. Regardless of the approach selected, communication of the overall plan and responsibilities needs to be clear so that System Implementation can be brought to a clean and clear end.

### Measurements of Success

System Implementation serves as its own Measurement of Success; indeed, a smooth System Implementation culminates – and validates – the entire system development effort.

Nevertheless, even before the final turnover, the Project Manager can utilize the measurement criteria below to assess how successfully the implementation is proceeding. More than one “No” answer indicates a serious risk to the success of this phase – and the entire project.

Figure 6-4

Process	Measurements of Success	Yes	No
Prepare for System Implementation	Has anyone verified that every Consumer has the right level of system access and security?		
	Is there a checklist of all system components that can be used to verify that all the right versions of all components of the system are in the production environment?		
	Do the managers of Technical Services and Technical Support agree with your estimate of extra work for their units associated with new system deployment?		
Deploy System	Do your team members agree that their part of the effort as outlined in the Project Implementation and Transition Plan is reasonable and achievable?		
	Do the training evaluation forms filled out by Consumers and Customers being trained in the new system reflect scores equal or higher to those anticipated in the Project Implementation and Transition Plan?		
	Have you had to “freeze” or “fall back” in system deployment activities no more than originally anticipated in the deployment plan?		
	Is the volume of support calls within the range originally anticipated in the deployment plan?		
Transition to Performing Organization	Has the Performing Organization agreed to transition all of the remaining defects along with the system itself?		

Phase Risks / Ways to Avoid Pitfalls

**PITFALL #1 – DAMN THE TORPEDOES, FULL SPEED AHEAD!**



Admiral David “Old Salamander” Farragut may have won the battle of Mobile Bay in 1864 with that command, but for a typical Project Manager, a planned “freeze point” should serve far better when the first mine explodes under the new system.

During the course of System Implementation, the Project Manager should have many points where the process can be frozen while the minesweepers fan out or a hole is patched. Think of it as a space shuttle countdown – NASA has frozen the clock with as little as 31 seconds before launch when the conditions warrant and a problem was discovered.

Having multiple pre-planned go/no go points during Implementation will spare you many grey hairs and sleepless nights.

**PITFALL #2 – ABANDON SHIP! WE MESSED UP PRODUCTION!**



The more extensive and complex the system, the better the chance that something will go wrong in production, no matter how well the System Acceptance phase went. That’s why it behooves the Project Manager not only to execute a comprehensive check of the entire production environment EVERY time anything is moved to production, but also to have an orderly fall-back plan for restoring production to a workable condition when – not if – something goes wrong.

Some error conditions are obvious and unmistakable – wrong heading on a screen, an improperly calculated total on a report; others are insidious in their perniciousness – a database update mechanism that deletes rows infrequently and at random, or miscalculates results by a small fraction. Those conditions may persist for days before being detected, and present insurmountable challenges in absence of a deliberate plan for retreat.

Save snapshots of the database; concatenate transactions; mothball but do not discard older versions of application code. Be ready to roll back or to jump back and roll forward – as long as you are not rolling off the deck of a sinking ship.

---

**PITFALL #3 – IT’S ALL MY FAULT!**

---



There is enough blame to go around in a typical System Implementation scenario. Something gets forgotten, something does not work right, something happens that is not planned for... and a good first step in fixing the problem is acknowledging the responsibility for it. However, sometimes the fault is not yours – even if someone is convinced it is.

Take, for example, a Consumer’s reaction to the new system, especially if said Consumer was not directly involved in System Acceptance. It is possible, indeed likely, that some feature of the new system will be at odds with what the Consumer thinks it ought to be. After all, it was requested by somebody else, and somebody else again built it. So a vocal Consumer will complain that the new system is wrong.

This situation is dangerous on two fronts: first, if accepted on its face value, it may mean rework, delays or worse; second, if not handled correctly, it may taint the perception of the new system and may lead to more complaints.

This is another case where solid, signed documentation really pays off. Prove that the functionality works just as it was requested. Enlist the help of the Project Sponsor, the Customer Decision-Makers, and any other “persuasive peers” (who are most likely just as anxious to have the system well received as you are), to explain the rationale behind design decisions and the process for change. And don’t accept any more blame than is properly yours.

---

**PITFALL #4 – THE IMPLEMENTATION THAT NEVER ENDS**

---



There is a song that never ends, it just goes on and on, my friends... at least until you stop singing it (probably because people are throwing things at you). But what if you are stuck in the Implementation phase that just does not seem to end? As soon as you fix all the technical problems, a Consumer reports a new bug, and you go through the cycle of fixing and testing and moving and checking, and then another Customer requests an urgent change, and the implementation cycle starts all over again, and then you encounter another technical issue, and it just goes on and on and on...

Somewhere – preferably sooner rather than later – you have to cut the cord, cross the Rubicon, make like a tree and leave well enough alone.

It certainly helps if you have a planned turnover procedure and a solid Project Transition Plan. It is also beneficial to have a Project Sponsor who understands the difference between system development and system support. And most of all, it is necessary to have courage and resolve to say “Basta!” “Finito!” and “Arrivederci!” (and not necessarily in Italian, either.)



### Frequently Asked Questions

#### **What is a pilot? How do we do it? How do we move out of the pilot phase?**

In system development parlance, a “pilot” is one of the techniques for deploying the system. Another technique is called “phased implementation,” although both terms are sometimes used interchangeably.

There are four main ways to roll out the new system. One – you can do it all at once, deploying all parts of the system to all the Consumers in one fell swoop. Two – you can deploy it piece-meal, releasing some of the system today, a little bit more tomorrow, taking it easy, rolling it out one part at a time. (That’s what we’ll call “phased implementation.”) Three – you can release the whole system in one shot, but only to a small group of friendly users. Once you verify that your test community survived the experience, you roll the system to another group, moving up the chain until you dare to expose your creation to your harshest critics. That’s what we’ll call a “pilot”. Finally, for very large and mission critical systems, you can do phased implementation in a pilot mode – roll out parts of the system to small groups.

Each approach has its pluses and minuses. Specifically for the pilot, the advantages are lessened exposure and an extra opportunity to test the system before releasing it to the world. The great disadvantages are having to maintain and coordinate two parallel processes, stretching out the deployment, and tying up Project Team resources for an extra long time.

If you do decide on a pilot, make sure the pilot group represents if not all, then at least the lion's share of business requirements. And once you document that the system can handle them adequately – move on. It is very important to limit your actions to addressing legitimate system bugs only; if you start tweaking and enhancing and adding functionality, you will never get out of the pilot phase.

**Does the system need to be perfect before deployment?**

“Perfect” according to whom? What may be acceptable for one Customer may not be good enough for another; and unless you involve your whole user base in accepting a fully functional system, you will not know whether your system is “perfect”; and by then – guess what – you’ve deployed it!

If you had a good representative cross-section of Customers and Consumers doing your System Acceptance, if you had cogent acceptance testing plans for them, if the Customer Decision-Makers signed off on acceptance test results, and if your team is confident in your deployment plan – by all means, take the leap, and release your baby into the wild.