

# Release go-Live day Preparation: A light Weighted Framework to Ensure Smooth Release

**Kalyan Thatavarti**

Xpedite Technologies Inc, 8830 Stanford blvd, Ste 314, Columbia, MD USA

**Abstract**— Now days IT software release go live day preparation is becoming more important in many organization that leverage IT as service. This is due to frequency of releases and limited availability of the installation window for growing day to day business needs. The purpose of this project is to explain the importance for software release go-live day preparation and to put the process in place to plan, monitor and control a scheduled release. The process framework is laid based on the study and analysis of field data of the IT software releases in top investment banks. The process is applied using Plan-Do-Check-Act, an iterative four-step problem-solving process typically used in process improvement on major releases across the bank. The aim of this paper is to establish an up to date over view of existing knowledge to benefit practice and future research. We show that some of the proposed benefits are realistic but that further research and improvements are needed to get the full potential value.

**Key words:**

Framework

## 1. Introduction

Many organizations develop the product by releasing the code frequently in an iterative manger on production environment. The Release go live day plan provides detailed information for subsequent planning, management, and step by step monitoring of sequence of activities that is coordinated with different teams. A release go live day plan also describes what portions of the system functionality will be implemented in which releases and the rationale for each release. The creation of the release plan should begin early in SDLC process and no later than Planning Phase, with a base lined version completed by the end of the Design Phase. The Release Plan should be updated incrementally each time the release goes into non-production environments and any time that the release strategy changes up until the system is fully functional in the production environment. The release plan becomes an input release go live day plan for better estimate of the outage time.

The base lined release plan is just a simple rolling wave planning having the sequence of activities with estimated duration of timing and the resource allotted for completion of activity for any integration environments. Fig 1.0 is the diagrammatic representation of base line release plan and is

incrementally updated for each intern release to QA, UAT and pre-prod environments.

## 2. Release Warm-up process:

Generally release warm up process starts at least a week before production go live date. This includes checking source code repository for any code changes, the applications code is compiled and built into binaries, data base package is prepared, checksum validations etc. In fact the primary responsibility of release management team is to make sure all the applications are at least tested in one of the non-prod environments. The application components that are tested in any non productions need to be staged and at least have two copies maintained on the release branch for contingency. The staged environments need to have enough read only access permissions for implementation teams.

### A. Release Readiness Criteria.

The Release Readiness Criteria is the final management checkpoint and approval stage before the release team begins detailed rollout planning and preparations. The Release Readiness Review results in a go/no-go decision about whether to deploy the release. If the decision is a go, the release moves to rollout planning and preparations or is physically incorporated into the production environment. Otherwise, the release is postponed until the necessary improvements take place, or it is canceled. Either way, the change log should be updated.

### B. Go/No-Go Meeting:

Go/No-Go is a meeting that is conducted with all the business users and stockholders is an extremely important milestone in release management process. This is just to make sure that everyone is truly ready and give you the thumbs up. The meeting increases the confidence levels to all the stakeholders based on their answer on firsthand analysis with total confidence that their application, team, or technical aspect of the activation is ready to go. Table 1

indicates go/no-go scenario based indicator for Go or No-Go.

### I. Installation Guide:

The Installation guide is most important documents that have the step-by-step process to install the applications programs in production environments. Installation documents can have download location of the application programs, Installation pre-requisites like Hardware/Operating system requirements, installation process, un-installation process, testing the installation. The technical documents is reviewed with all the development teams before it's handed over to implementation team. It's the responsibility of the release management team to make sure implementation team understands steps in the installation documents and signs off the documents before the implementation.

The location of the installation guide is placed either in shared location like share point or local wiki with read only access permissions. This is normally updated by release management team as and when required. Any unauthorized changes to this document will put the release at risk. So release management team should make sure there is only single point of contact and ultimate source truth. Installation guide is normally owned by Release management team.

### II. Release Notes:

Just like Installation guide release notes is owned by release management team and is a communication documents shared with customers and clients of an organization detailing the changes or enhancement made to the features of service or product the company provides. Thus this communication document is usually circulated only after the product or service is thoroughly tested and approved against the specification provided by the development team. Release notes are frequently written in the present tense and provide information that is clear, correct, and complete.

Release notes doesn't have standard format that is followed throughout different organizations. Organizations normally adopt their own formatting styles based on the requirement and type of the information to be circulated. The release notes normally have the release number, release date, type of the fix, new functionality, defect fix, the component, component type, version component/business rules impact. The content of release notes also vary according to the release type. For products that are at testing stage and that are newly released, the content is usually more descriptive compared to release notes for bug fixes and feature enhancements, which are usually brief.

### III. Transition Plan:

The Operations and Maintenance Transition Plan is designed to facilitate migration of an application system from a development environment to a production / maintenance environment. This plan is to train the supporting teams for day to day operations of the production support. This includes regular environment health checks like checking logs, monitoring the errors/exception in the logs, cleaning the logs, & monitoring the EOD/Month end jobs. The transition plan shall also define management controls and reporting procedures, as well as the risks and contingencies. The plan includes known issues or problems & if problems arise, how to identify and report. If necessary, temporary "work-around(s)" is defined.

### IV. Implementation walkthrough meeting:

In order to make discussion of the implementation more concrete, a meeting is conducted among the stakeholders to brief out the release plan and to walk through the implementation plan. The meeting is to agree and discuss about their responsibilities and to make then understand their task for better coordination on go live day. The stakeholder can be the production support team, development teams, database administrators, networking and other infrastructure teams.

**Meeting Purpose and Goals:** Release Management team drives this meeting to better command & control the release. The main objective is as follows.

1. To help cross-team communication.
2. To help understand their responsibility.
3. To better coordinate among the teams.
4. To better understand who is doing what and to better estimate and agree the timelines.
5. To bridge the gap between deployment and implementation teams.
6. To identify potential areas of risk and to mitigate the risk.
7. To better understand command and control of instruction from release management team.
8. To identify primary and secondary resources from each individual team.

- 9. To identify first line of defense and other escalation procedures.
- 10. To identify who is on call, on-site, offshore and off site etc.

**Guidelines for On-Site, On-Call, Off-shore, Off-site:**

**On-Site:** The person need to physically report in the command center in the office where the release is being implementation as defined in the release plan.

**On-Call:** The person should be available with two hours of radius distance and should be able to answer the call.

**Off-Shore:** The person can be physically located in any country and should be readily answerable to emails and conference.

**Off-Site:** The person should be can be anywhere within the country and should be available to answer the calls.

**Guidelines for Primary and Secondary resource from each team:**

**Primary:** First line of defense for all issues related to team and should be On-Site during the release implementation.

**Secondary:** Second Line of defense, in case the primary resource cannot handle the issue and can be On-Call during the release implementation.

Levels of Escalation	Name of the person	Purpose of Escalation
First Level	Name1(any from RM team)	Updates, Issues, Conflicts
Second Level	Name1 & Name2(Lead)	Needs more outage time
Third Level	Name1 & 2, Management	Rollback the release

**Table 1.0**

**a. Release Communications:**

There are two types of communications from release management team.

- a) **Outage communications:** Communicate to all the stakeholders informing about the outage of the environments, description, purpose, applications impacted due to this outage and unavailability timings. The unavailability timing needs to be indicated as per their geographical locations. This communication need to be through email.
- b) **Release Updates:** Communicate the entire stakeholder/Management about the progress of the release and issues raised and fixed throughout the

implementation. This communication can be either by email or conference update.

- c) **Delegation and control:** Communications to all the teams involved in implementation to better coordinate and control the entire release process. This communication can be through email or through conference.

**b. Testing & sign off approvals:**

Testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. It's the most important phase of the software development lifecycle that can be relied by all the stakeholders and release management teams to make a go/no-go decision for production rollout. It's the duty of the release management team to see that code is well tested and signed off by quality assurance and business users before rolling out to production. The sign off approval can be either by email or a word document attached stating what is tested in which environments on what data. The signoff approval should be from business users and from quality assurance team.

**User Acceptance Testing (UAT):** User Acceptance test is normally conducted by business. Business users should focus on testing a set of requirements (that is expected to move to production as a Production Release) to ensure that the release meets user expectations.UAT testing should certify that the requirements meet user expectations and this is a mandatory signoff.

**Quality Assurance testing (QA):** Quality Assurance testing is normally done by quality engineer and main focus on previously agreed specifications, standards and functionality required without defects and possible problems. It monitors and tries to improve the development process from the beginning of the project to ensure better quality and is mandatory approval.

**b. Go Live Day:**

Normally go live date is planned on Saturday of any week, to make enough Contingency buffers to mitigate any issues during rollout process. The buffer is added to the release plan for any extra time needed for issues, business readiness test (BRT) and other rollback strategies. So at the worst case the rollout may be extended till close of Sunday end of the day. It's the responsibility of the release management team to make sure the production environments is open, up and running for business by Monday morning. The baseline is all the release activities that are planned should not interrupt business continuity.

### 3. Sign-In:

Release Management team coordinates all the operations of the release through command center on go-live day. Command center is conference room that has all the electronic equipment necessary to open the bridge call. The bridge call is a dial in number with conference participation number. The implementation team who are on site should report to command center as mentioned and agreed in the release plan. All other participants can dial in to the conference. If there is any delay in reporting, need to be informed command center. Command and control flows from top to bottom and inform is flows from bottom to top. It is recommended that all the installations should start by 6:00AM on Saturday.

### 4. Installation plan

#### Shutdown all the applications:

- a. **DB Backup:** All the installation start with Hot/Cold data base backup. The replication between production database and business continuity server (BCP) is taken down. Make sure the backups are stored on the tap for restore in case if there is rollback of the release.
- b. **Shutdown Web Application:** All the front end application servers are stopped and copy of the ear, war, jar files are backed up and placed on the storage tape for contingency.
- c. **Shutdown all standalone applications:** All Java base standalone applications (Non-Web base applications) should be shut down.
- d. **Shutdown all the schedulers:** All the schedulers are stopped to make sure no jobs/processes are running during the time of installation.
- e. **Shutdown all the services:** All the services that are running for database need to be shut down.
- f. **Shutdown ETL bridges/components:** Insert down Times for all ETL related components to stop all the jobs running in the production environments.

#### Start Installation:

- a. **Upgrade the database:** Once the all the backups are complete and all the applications are shutdown then the actual installations start. Database is the most important component as data is vital for business. Database install kit is run on database schema for all the database changes. The install kit log is constantly monitored for errors/exceptions.
- b. **Install Web Applications:** All the frontend/middle tier web applications are installed.

During the process of installation of application on all the logs are monitored for errors/exceptions.

- c. **Configure Configuration Files:** Install/upgrade all the configuration files for java standalone applications. These changes can be log level setting, variables/value locations. The older version of the configuration files is copied to an archive location before installing the configuration files.
- d. **Installation Standalone Applications:** Upgrade all the standalone applications that run independently like Master File engine, broadcast engine. These components need to be installed and should not be started.
- e. **Install Schedulers:** All end of day jobs, month end jobs that are run through the schedulers need to be installed.
- f. **Configure ETL bridges:** Configure all the ETL bridges taking one at a time. These ETL bridges feed huge volumes of data to the downstream systems.

#### Start All Applications:

- a. **Start up Database:** First components that should be started are the database instances.
- b. **Start up Web Applications:** The second components that need to be started are web applications that look for database instances.
- c. **Start up Standalone Applications:** Next are the standalones that run independently that broadcast the data from the database.
- d. **Start DB Replication:** Start the replication between production database and the business contingency server (BCP).
- e. **Start Scheduler/ETL bridges:** Next start all ETL and other scheduler jobs that are shut down during the installation process.
- f. **Start up All services:** Next start all OS services that are shut down during the installation process.

### 5. Business Readiness Test (BRT):

Once all the applications are up and running, it's the time test the application for business readiness. Basically the business readiness test is conducted after upgrade to production environment and before the production is handover to business on first thing on Monday morning. This is test which is conducted either business or any person from quality assurance team. This is one of the most significant event to say the all the installation is correct and all the applications are back up and running.

Ideally this test needs to be conducted at different locations by different people sitting across the globe. Business readiness test doesn't create any data in production data;

instead the functionality is tested at the high-level. This tells the release team whether to rollback the release or not.

## 6. Business Readiness Go/No-Go:

This is an open call that is conducted after business readiness test is done to take a decision whether or not to rollback or move forward with the release. If the BRT passes, the release is live for the business. If the BRT fails, then based on the component and the impact the release is rolled back. This call is conducted by RM team and all the implementers, BRT tester, Business and other higher management will be involved in the decision.

## 7. Rollback Strategy:

Rollback in terms of software release is uninstalling and committing back to last good known state. The last good known state of the production environments is that state before the installation. There are two main reasons to rollback the installation. Either the total installation is going out of window or the upgrade has functional defects where the business cannot live with it. The decision to whether rollback or not can be taken during business readiness go/no-go meeting.

- c. *Rollback plan:* The rollback plan is designed well in advance before along with release plan and the rollback plan is designed for each and every component. Without the rollback plan the release is not scheduled for production. This is a mandatory plan that is designed for the entire release with the help of the development teams. Impact analysis is done for all the rollbacks and is documented in the installation guide.

- d. *Escalations plan:* Along with the rollback plan, the escalation plan is designed and concern service/project managers are involved. Release team is responsible to design the Responsible, Accountable and inform the concern stakeholders/business.

## 8. Postmortem of the release/Lessons learnt:

Once the release is rolled out to the production, release team documents all the issues and does the root cause analysis for the issues. This is a mandatory activity that release team should do for the better planning of the upcoming releases. The root cause is done for each and every issue and triage items are raised to the concern persons for the improvements. If the issues are raised against release team, the team needs to conduct a retrospect meeting and discuss about the release issues. All the

activities starting from planning till postmortem of the release should be documented.

## 9. Process Improvement:

“Process improvement” means making things better, not just fighting fires or managing crises. It means setting aside the customary practice of blaming people for problems or failures. It is a way of looking at how we can do our work better. In process improvement, we seek to learn what causes things to happen in a release process and to use this knowledge to reduce variation, remove activities that are no value release, and improve rollout plan.

A simple Plan-Do-Check-Act (PDCA) Cycle Fig 1.1 is used iteratively on the release plan to constantly improve the existing process for better estimate the release window.

## Conclusion:

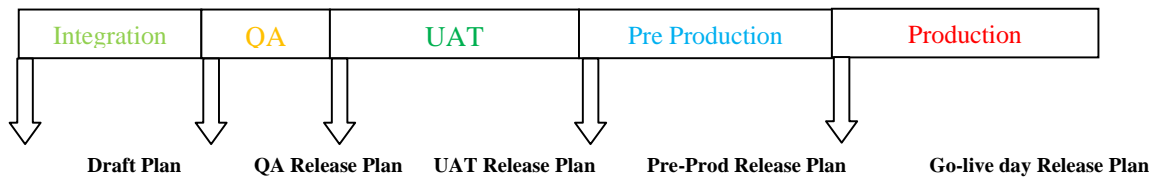
### Results 1. Controlled Release

#### 2. Stick to scheduled outage window.

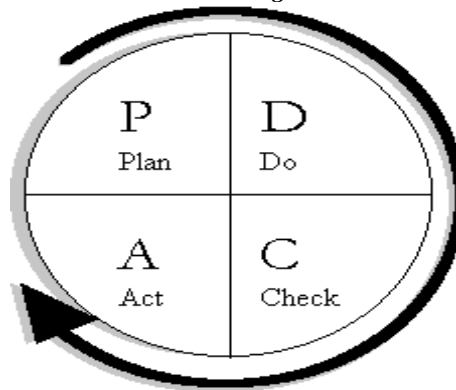
### References:

- [1] <http://technet.microsoft.com/en-us/library/cc535326.aspx>
- [2] <http://www.hci.com.au/hcisite3/toolkit/pdca/cycl.htm>
- [3] [http://technet.microsoft.com/en-us/library/cc736773\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc736773(WS.10).aspx)
- [4] R. Conradi and B. Westfechtel, “Version Models for Software Configuration Management,” ACM Computing Surveys, vol. 30, no. 2, 1998.
- [5] ANSI/IEEE Std. 1042-1987, IEEE Guide to Software Configuration Management Content, IEEE Press, Piscataway, N.J., 1987.
- [6] Brady Orand, “Foundations of IT Service Management: The Unofficial ITIL V3 Foundations Course”.
- [7] Randy A. Steinberg “Architecting ITIL: A Reference for Architecting the Complete Enterprise Architecture and Configuration Items Needed to Operate an It Service Management Infrastructure”
- [8] Kevin A Lee, IBM Rational(R) ClearCase(R), Ant, and CruiseControl Mill Valley, CA: University Science, 1989.

**Incremental updates to Release Plan in Integration and Testing Phase of SDLC:**



*Fig 1.0*



*Fig 1.1* Plan-Do-Check-Act (PDCA) Cycle

Element	"Go" indicators	"No-Go" indicators
Release	<ul style="list-style-type: none"> <li>All prerequisites have been met.</li> <li>All documentation is in order.</li> <li>Adequate rollback plan exists.</li> </ul>	<ul style="list-style-type: none"> <li>Release is not built to standards.</li> <li>Documentation is missing.</li> <li>Primary and secondary support personnel have not been assigned.</li> </ul>
Production environment	<ul style="list-style-type: none"> <li>Support staff is already trained.</li> <li>All administrative procedures for the release are clear and aligned with site standards.</li> <li>Operating level agreements (OLAs) and underpinning contracts (UCs) are in place and appear to support required service levels.</li> </ul>	<ul style="list-style-type: none"> <li>Software levels used to build the solution are unsupported (too new, too old) in the current environment.</li> <li>No support staff training plan.</li> <li>No backup plan.</li> </ul>
Business/ Stakeholders	<ul style="list-style-type: none"> <li>Business agrees that all the functionality that is newly developed has met adequacy.</li> <li>All key stakeholders have been notified.</li> <li>Business continuity plan/Disaster recovery plan exists.</li> <li>All QA testing is done and signed off</li> </ul>	<ul style="list-style-type: none"> <li>Newly developed functionality doesn't meet the business requirements.</li> <li>Plan prerequisites have not been met, such as confirmation that users will be on hand to do acceptance testing.</li> <li>Change management documentation is not in order.</li> <li>Poor quality poor standards.</li> </ul>

**TABLE 1** Go/No-Go Indicators