**DIEGO : Como vamos a determinar en Octopush , cuales son los rls que van a ser automáticos y cuales son los que están en la cola para que haga el deploy RM ?**

**DIEGO .. va a usar Otopush o vendemos que vamos a seguir con la ventaja de deploy parcializado.**

<https://confluence.olx.com/display/QA/Release+Management+Process+New>

This section defines the process associated with the management of deploying releases. The process cover the transfer of a new or enhanced product features to the production environment, provide tools and support to prevent disruption of existing services.

This will give us a unique standard process to be implemented in any OLX team with the aim to increase the number of successful releases, including reducing the number of releases with unexpected outcomes, decrease the number of incidents caused by releases and reduce the leap time to market.

This section defines the process associated with Release Management. RM is the process responsible for planning, scheduling, and controlling the build and deploying Releases. Release Management Team provide support and tools for  delivers new and enhanced product features required by the business, while protecting the integrity of existing services

This will give us a unique standard process to be implemented in any OLX team with the aim to increase the number of successful Releases, including reducing the number of Releases with unexpected outcomes and decrease the number of incidents caused by Releases.

<https://confluence.olx.com/display/QA/PCS+-+Release+Management>

# Purpose

The primary goal of the Release Management Process is to increase the number of successful Releases, including reducing the number of Releases with unexpected outcomes**,** decrease the number of incidents caused by Releases **and reduce the leap time to market.**

# Scope

This document should be used by OLX Teams:

* Involved in request changes in our OLX Product.
* Involved in the operation and management of Release Process
* Responsible for the restoration of product features
* Involved in incident or bug management.

Any Release provided by Suppliers are excluded of this initial version. Development and testing environment requests are not part of this process either, however some of agreements of this process could be applied.

# Policies

* **Release Point of Contact** - A single “Release PoC – Point of Contact” must be identified in every Team/Component/Project.  The Release PoC will be responsible for successful coordination and execution of the Release Process, as well as ensuring all required information related to the RLS is provided. (See [*Release Point of Contact Page*](https://confluence.olx.com/display/QA/Release+Point+of+Contact))
* **Major Release Plan** -  Program Coordinator and Release Manager must define and communicate the weekly release Plan based only on Major Release for the specific week. It is expected that all Major Release should be completed by the Planned End Date/Time defined. (See *[Major Release Plan Page](https://confluence.olx.com/display/SM/Tentative+Release+Weekly+Agenda)*)
* **Queue Mechanism** - All RLS requests that are not part of the Major Release Plan is managed using a Queue, the Queue will be manage using the FIFO concept where every team is able to create RLS request on demand.
* **RLS Visibility** - RM team provide a JIRA board to all teams to see the progress of each RLS request as well as the RLS Queue. (See [RLS Board](https://jira.olx.com/secure/RapidBoard.jspa?rapidView=114))
* **Emergency Criteria** - Critical Bug and Incidents are the only issues considered as a standard emergency release (Hotfix), they would affect RLS Queue. RM Team will received a notification when a Critical bug or Incident is created. Exceptions will required CTO authorization. (See*[Exception Criteria](https://confluence.olx.com/display/QA/PCS+-+Release+Management" \l "PCS-ReleaseManagement-EC)*)
* **Manage Changes** - Every Day the RM Team verify the RLS Queue communicating every changes to the team involved. Changes in Major Release Plan are controlled and managed by Program Coordinator. Urgencies and Unplanned activities that impact the RLS Queue are managed in a daily basis.
* **Windows Timeframe** - Release PoC and RM team will use the Windows Timeframe defined for coordination and execution of RLS request. Minor Release requested out-of-windows timeframe defined as well as non-planned Major Release will required authorization and will be treated as an exception. (See [*Release Windows Timeframe*](https://confluence.olx.com/display/QA/PCS+-+Release+Management#PCS-ReleaseManagement-RWT))
* **Release Support** - Every execution of a RLS requested should be attended by the RM (or any replacement if he is absent for any reason), a SysOp from Infrastructure Team, a DEV resource responsible for the deployed features and a designated team member for testing as soon as the RLS is LIVE. This not applicant for the automatic release. (see automatic release police)
* **Feature Freeze** - Feature freeze mean that the team are not able to do merge on Master Branch. Just Bug Fixing is allow. On feature freeze RM Team will blocked staying environment ~~testing env.~~ every Release (two weeks) on Monday at 12:00pm to Wednesday at 10:00am. Every teams must ensure that this integration doesn't affect their product. (See [*Release Feature Freeze*](https://confluence.olx.com/display/QA/PCS+-+Release+Management#PCS-ReleaseManagement-RFF))
* **Queue stop** – On specific condition like critical incidents, massive disruption, low performance, critical change, the RM could stop the automatic release queue to prevent more damage or noise in the production environment and perform a better root cause analysis identification of active incident.
* **Automatic release** – Allow to deploy directly to production environment without intervention, support o monitoring from a central release team management OR TIME FRAME WINDOWS CONSIDERATION.
* **Formal Channels** - The formal channels that must be used are JIRA, #release IRC channel as well as [release@olx.com](mailto:release@olx.com) email address.

## Detail Description

#### RLS Mandatory Information

* Versions and Deploy Steps, SQL Scripts to Upload.
* ROLLBACK steps and Rollback scripts attached.
* ~~For automatic release the mandatory information is …..~~
* ~~Automatic release DIEGO que ponemos ?~~
* Asegurarse que el tK es “correcto”
* Tk de cambio asociuado
* Red to live
* Close -.. REVISAR DONDE ESTA ESTO.

#### Exception Criteria

* Critical Bug and App Incidents are the only issues considered as a standard emergency release.
* Minor Release requested  out-of-windows timeframe defined as well as non-planned Major Release will required CTO authorization and will be treated as an exception.
* Automatic release not allow exception criteria. There not restriction that can be jump for a CTO authorization. If some authorization or approbation is necessary, each market have to establish the better practice for these concept.
* STOP DE COLA !!!!!!!!!!!!!!!

All exception requested will be tracked and measured to analyze root cause and define action plan to reduce those situations (Action Plan will be part of the our continuous improvement process mechanism - see 4.1.1 Metrics )

#### Release Priorities

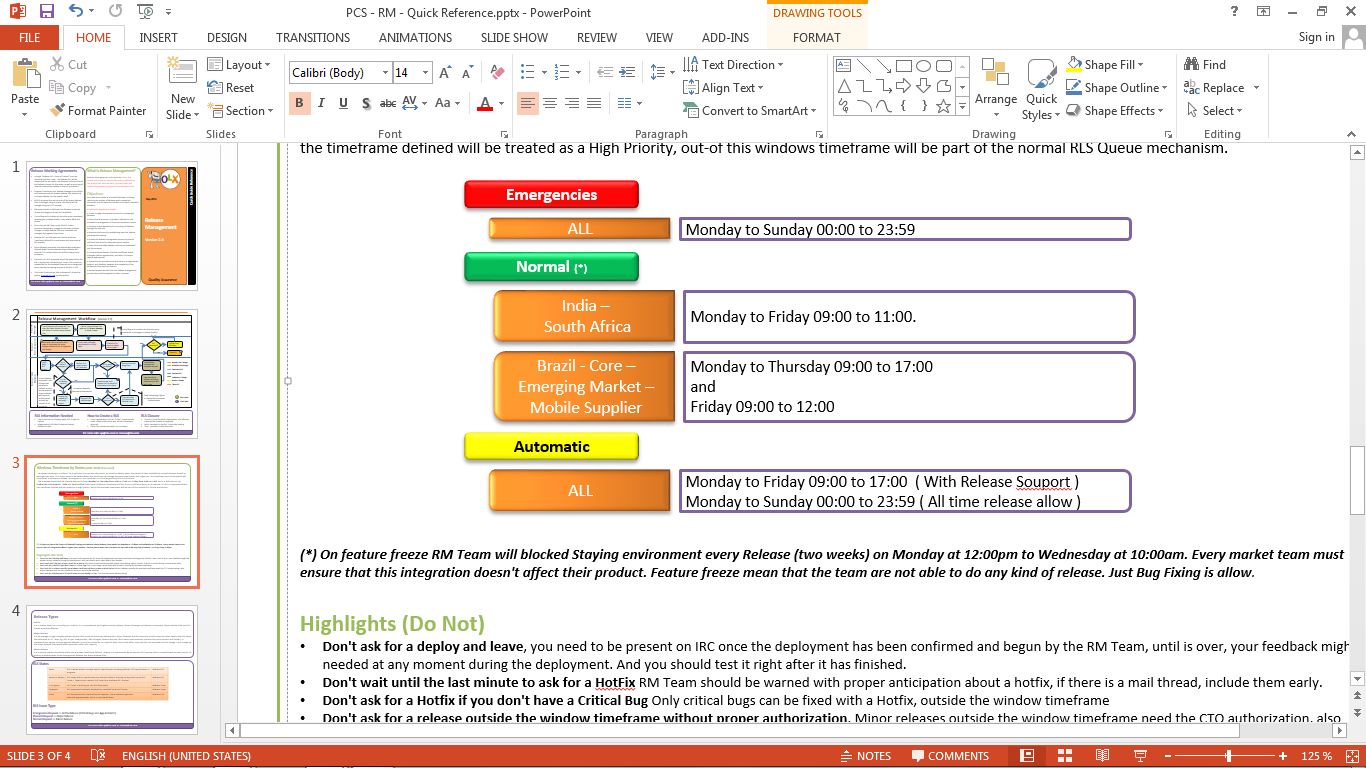
* **HotFix (Emergency Request)**: They are non-planned and highest priority releases. When a HotFix is requested, Major Release Plan and RLS Queue would be affected.
* **Major Release (Planned Request):**They represent the highest priority planned release. They must be scheduled to a specific date due to the effort required and risk associated to those changes. Every change on Major Release Plan would affect priorities, effort and capacity.
* **Minor Release (Normal Request):** Represent on-demand RLS requests that are created based on teams needs. They are subordinated to the Emergencies Release and Major Release Plan.
* **Automatic Release:** Represent the way to go directly to production environment, have the same priority of minor release.

#### Release Windows Timeframe

Windows timeframe is defined  for Production Environment Requests, to provide visibility about the period of time available for normal releases as well as emergencies ones. OLX teams need to be aware about this timeframe to manage business expectation and urgencies. This timeframe have to be specifically considered in automatic release, the support in this timeframe is crucial to guaranty the SLA accorded.

RM windows timeframe for Normal Request is from **Monday to Thursday from 9:00 to 17:00** and**Friday from 9:00 to 12:00**, Minor Release requested out-of-this-windows timeframe defined as well as non-planned Major Release will required authorization and will be treated as an exception.

**India** and **South Africa** Team have a different timeframe but this is just to define priority on its request, if a RLS is requested within the timeframe defined will be treated as a High Priority. All RLS requested  out-of this windows timeframe will be part of the normal RLS Queue.



#### Release Feature Freeze

Every Release Cycle (two weeks) a full release is made on our framework, and for that there is strict a timeline that has to be followed: The beginning and the end of the Release is announced by e-mail to[**tecnology@olx.com**](mailto:tecnology@olx.com), and all communication during the Release is made thru the #release IRC channel.

Pedimos que nos agenden / Avisen cuando estiman realizar los releases automáticos ? o los independizamos totalmente ? Si se independizan perdemos control (desconocimiento de que están haciendo) pero ganamos en independencia

* **Monday's noon**: You will receive an email from the RM letting you know about this, which means that no new features are to be committed and staying environment is about to be blocked. Only emergencies issues are accepted (critical bugs and app incidents).
* **Tuesday day**: Every team must ensure that the integration of our platform doesn't have any issue related, that means that  all teams need to start immediate a regression test in testing environment. If something is wrong Release PoC must contact RM team to review it and resolved the issue inmediately.
* **Wednesdays 10:00AM**: The full release to PROD is made and staying environment is unblocked, the RM, a DBA and a SysOp are always present during this process, but there also should always be DEV team members ready to test as soon as this is ONLINE

#### Release Workflow Status

Workflow is the movement (or transition) of an RLS through various Statuses during its lifecycle. The following diagram shows OLX Bugs workflow, where:

* **Open**: RLS request waiting to be picked up by RLS Team – Teams must contact RLS Team thru #release IRC channel.
* **In Progress**: RLS Team is working on the RLS Request.
* **Deployed**: RLS request is already deployed or resolved.
* **Done**: RLS Request was checked and is considered done.

## Roles and Responsibilities

* **Product Team**:
  + Work with Release PoC every week to determine the Major Release Schedule focus on Urgencies and Priorities.
* **Technology Team**:
  + Work with Release PoC every week to determine the Major Release Schedule focus on Urgencies and Priorities.
* **Release Point of Contact (PoC)**:
  + Work with Technical and Product team every week to determine the Major Release Schedule focus on Urgencies and Priorities.
  + Verify ready-to-deploy requirements. When a Release is required, create RLS ticket using our Elvira mechanism.
  + Request the approval to the CTO for exceptional releases.
  + Make the automatic release to production environment.
  + Do the necessary monitoring an evaluation to ensure the stability and functionality of production environment after automatic release.
  + Make the automatic rollback for release that produce disruption to production environment.
* **Program Manager**:
  + Coordinate with all Market and Core team the tentative Major Release Schedule and define a common Major Release Plan.
  + Coordinate the Major Release Plan with the Release Manager to understand dependencies, effort needed and capacity affected.
* **Release Manger**:
  + Negotiate on the scope of the Major Release Plan.
  + **RM O OPERACIONES ???**Stop the queue in critical incident, feature freeze, know functional dependencies or critical change.
* **Release Team**:
  + Verify the RLS Queue and analyze changes
  + ~~Request the approval to the CTO for exceptional releases~~
  + Guaranty the approval for exceptional releases.
  + Implement the Change
* **CTO**:
  + Approve or reject the exceptional release

# Metrics

Metrics reports should generally be produced bi-weekly with quarterly summaries. Some metrics to be reported are:

| **Metric** | **Purpose** |
| --- | --- |
| Release Quality  (by team and Release type)   * Amount of RLS * Cumulative RLS * RLS With Rollback * Successfull vs Unsuccessfull | To determine if release techniques and processesare  well implemented and followed, and to detect  improvement process areas. |
| Process analysis  (by team and Release type)   * RLS resolution lead time * RLS answer Time * Emergencies RLS    + Critical Bugs   + App Incidents * Exception RLS   + Authorized Request (Approved by CTO) | To determine if RLS are being processed in a  reasonable time frame, frequency of specific  types of RLS, and determine where bottlenecks exist. |

# Document Related