ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU2A004

ENSURING SMOOTH CONTROLS UPGRADES DURING OPERATION

M. Gourber-Pace†, F. Hoguin, E. Matli, B. Urbaniec, W. Sliwinski, CERN, Geneva, Switzerland

Abstract

The CERN Accelerator Controls systems have to remain as stable as possible for operations. However, there are inevitable needs to introduce changes to provide new functionalities and conduct important consolidation activities. To deal with this, a formal procedure and approval process, the Smooth Upgrades procedure, was introduced and refined over a number of years. This involves declaring foreseen Controls changes as a function of the accelerator schedules, validating them with stakeholders, and organising their deployment in the production environment. All of this with the aim of minimising the impact on accelerator operation. The scope of this activity is CERN-wide, covering changes developed by all CERN units involved in Controls and encompassing the whole CERN accelerator and facility complex. In 2022, the mandate was further extended with a more formal approach to coordinate changes of the software interfaces of the devices running on front-end computers, which form a critical part of the smooth deployment process. Today, Smooth Upgrades are considered a key contributor to the performance and stability of the CERN Control system.

This paper describes the Smooth Upgrades procedure and the underlying processes and tools such as schedule management, change management, and the monitoring of device usage. The paper also includes the major evolutions which allowed the current level of maturity and efficiency to be reached. Ideas for future improvements will also be covered.

INTRODUCTION

Making controls changes during a beam run is desirable to deliver novel and enhanced functionality as requested by the CERN Operations team. It is a delicate procedure necessitating meticulous preparation and execution to ensure the preservation of the accelerator performance and stability. This paper describes the formal process applied at CERN, to prepare and execute the deployment of controls changes during beam operation. An emphasis is placed on the importance of documentation, approval, and communication along the process to mitigate adverse effects on operations.

BACKGROUND

As part of the annual official CERN accelerator planning, a number of beam stops are scheduled to facilitate necessary upgrades and maintenance interventions aimed at enhancing the performance and reliability of the accelerator complex. These stops can be classified into two types:

- **A Technical Stop (TS),** which occurs once or twice a year and typically lasts between 12 to 24 hours.
- **A Year-End Technical Stop (YETS),** which spans several weeks starting in November or December.

Both the TS and YETS periods provide an opportunity for implementing and deploying controls upgrades. During these intervals, accelerators are stopped to facilitate various interventions on components such as radio frequency cavities, magnet power supplies, beam instrumentation, etc. Across the entire accelerator complex, an established protocol dictates that controls upgrades should only be executed during a TS or YETS. The only rare exceptions may be a bug fix or specific new feature deployment urgently requested by Operations teams.

This policy has grown more rigorous over time, stemming from lessons learned from past controls upgrades. In 2015 and 2016, a qualitative assessment was conducted, uncovering that controls upgrades had adverse effects on LHC performance. Following controls software deployments, numerous hours of beam operation were compromised. Several factors contributed to issues during the restart after TS and YETS events, including: inadequate pre-deployment testing, underestimation of the impact on interconnected systems, introduction of non-backward compatible changes, insufficient communication resulting in Operations' unfamiliarity with the deployed changes.

Drawing on this experience, the Smooth Upgrades procedure was formulated to facilitate a coordinated deployment of controls changes across the accelerator complex.

SMOOTH UPGRADES PROCEDURE

Mandate

The Smooth Upgrades (SU) procedure outlines a method to be applied during all TS and YETS periods, aiming to streamline deployments and minimize the risk of impact on accelerator operation. The SU procedure covers several interconnected needs:

- **Central Repository**: to document planned upgrades from controls teams in a centralized repository. The goal is to compile a comprehensive list of changes, allowing the Operations team to correlate issues during beam restarts with recently deployed modifications.
- **Approval and Compliance Workflow**: to ensure that upgrades and interface modifications have received approval from the Operations team before proceeding with the implementation.
- **Conflict Analysis and Mitigation**: to identify potential conflicts amongst the planned upgrades and establish priorities and / or deployment order.
- **Risk Evaluation:** to assess operational risks, ensure the existence of validation procedures, and establish contingency procedures for rolling back changes if necessary.
- **Process Review:** to evaluate the effectiveness of the procedure after each TS or YETS, gather user

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU2A004

feedback, and define enhancements to both the workflow and the tools supporting this process.

Scope

The SU scope spans three significant dimensions:

1. The number of accelerators.
2. The quantity of control components.
3. The multitude of development teams involved.

Firstly, the SU procedure is applied to the entire CERN accelerator and experiments complex, which comprises 12 facilities.

Secondly, the SU procedure needs to cater for changes in all software elements within the controls domain, including:

- ~ 1,200 FESA [1] device classes
- ~ 100,000 FESA [1] device instances
- ~ 2,000 controls front-end computers
- ~ 1,000 GUI applications used by the Operations team

Thirdly, the SU procedure reflects that the aforementioned software components are developed and maintained by eleven distinct groups distributed across four departments within the CERN Accelerator & Technology Sector. Three centralized groups provide various controls frameworks and infrastructure (highlighted in light green background in Fig. 1), e.g. real-time frameworks, settings management, communication services, etc. These frameworks are then used by eight affiliated partner groups to create equipment specific controls (highlighted in light yellow background in Fig. 1), e.g. for kicker magnets, beam loss monitors, radio frequency, etc..

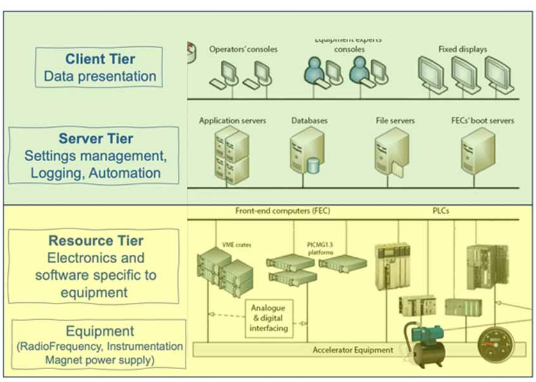

Figure 1: Controls architecture and responsibilities.

Boundary Conditions

The SU procedure adheres to stringent boundary conditions. In practice, TS and YETS are not complete stops of all facilities without particle beams. During these periods, certain facilities that operate outside of the main LHC accelerator chain, may continue their regular operation to serve physics experiments independently of the general TS and YETS schedule. Consequently, these facilities require operational and stable controls throughout the entire duration of a TS or YETS. This constraint entails the necessity to negotiate deployment time slots and strive for minimal impact.

A TS is only scheduled once or twice a year and typically has a duration of 30 hours or less. In practice, this duration is further condensed when it comes to controls upgrades, which typically need to take place within a limited time window of 8 to 12 hours. This is because operational controls are needed at the beginning and end of a TS to allow Operations to remotely activate and deactivate various equipment.

Organization

The SU team consists of members from diverse controls services, including the central controls groups and the equipment-specific controls teams. As the primary users of the control services, Operations team members complete the SU team. The membership consists of ~ 38 individuals, with 20 coming from the controls groups, 10 from equipment groups, and 8 from Operations. This team operates under the guidance of the SU Coordinator.

SU STEP BY STEP PROCESS

The SU process encompasses the following steps and involves various stakeholders:

1. **SU coordinator**: initiates the SU procedure by announcing the commencement of the process for the declaration of the controls changes foreseen for the upcoming stop (e.g. a TS or a YETS). S/he establishes a deadline for the approval of all controls changes and communicates key boundary conditions, such as the time window for declaration of controls changes, subsequent deployment, and the operational status of all facilities during the stop.
2. **Developers responsible for controls changes**: declare all proposed changes.
3. **Authorized persons from Operations with approval rights**: hold the authority to approve or reject the proposed controls changes.
4. **SU coordinator:** oversees the validation process and arranges a meeting when the deadline for controls change declarations has passed. This meeting involves key stakeholders, including controls change developers, operations representatives, and potentially experts from core controls groups impacted by the declared controls changes. The meeting's objective is to finalize the list of controls changes, establish their deployment order based on inter-dependencies, address potential risks and impacts on the operational environment, obtain final approvals, incorporate validation time slots into the TS/YETS schedule, and negotiate deployment slots for facilities that remain operational during the stop.
5. **Developers responsible for controls changes**: execute the deployment of their controls changes during the TS/YETS, adhering to the predefined sequence. They also keep Operations informed about the progress and status of the deployments.
6. **SU coordinator**: conducts a debriefing meeting in the days following the stop to review the deployment's success and identify any areas for enhancement.

Throughout this process, collaboration and communication among the SU team, developers, and Operations teams

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU2A004

are crucial to ensure a smooth and effective controls upgrade procedure.

SOFTWARE FOR SU PROCESS

Functionalities

A user-friendly web-based SU application facilitates the SU process. It covers the key stages of the controls change lifecycle, namely: declaration, approval, and deployment. Key attributes of the Graphical User Interface (CUI) are that it is easy to navigate and enable users to swiftly declare new controls changes in just a few clicks. The GUI is comprised of three main views:

7. **EDIT view** (example in Fig. 2**):** facilitates the creation and editing of new entries, resulting in a complete description of a controls change.
8. **LIST view** (example in Fig. 3): presents the option to display all changes or apply filters such as the targeted facility, the group responsible, the targeted TS or YETS, etc.
9. **HISTORY view**: provides a comprehensive record of all actions that have been carried out.

Advanced features to facilitate the edition are available, like the ability to clone entries and perform bulk updates.

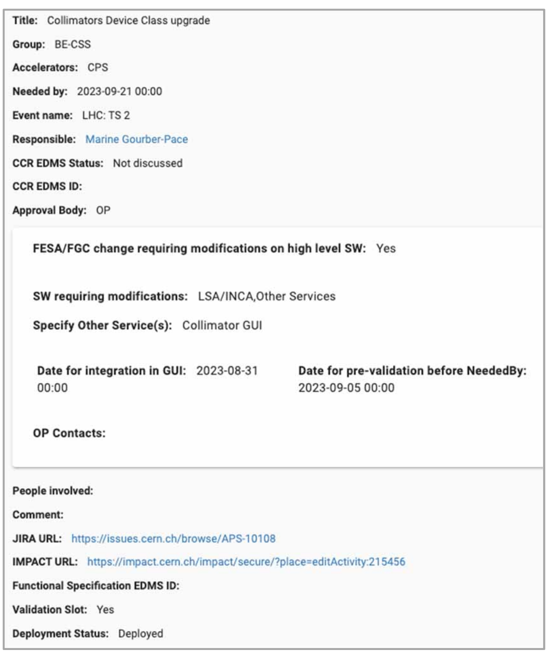

Figure 2: SU GUI “EDIT” view.

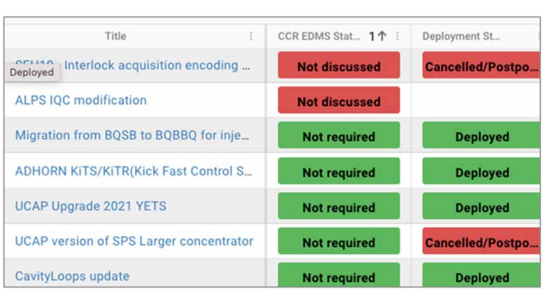

Figure 3: SU GUI “LIST” view.

Enforcing the SU Workflow

Several aspects of the SU application help enforce the well-defined SU workflow. For example, a change requiring approval appears in red in the GUI until approved. Approval is enabled for authorized persons only and an automatic notification will be issued to warn about the required approval ahead of approaching deadlines. Registering the deployment of a controls change is also forbidden if it has not yet been approved.

Integration With Other Tools

To avoid a duplication of effort with respect to other tools used at CERN, the SU application incorporates numerous attributes in a central place, that enable access to additional external information associated with the controls change declarations:

- **URL of related JIRA issue:** JIRA is typically used within CERN's control groups to detail development of functionalities in software products. As such, each controls change should refer to an existing JIRA issue for in-depth follow-up.
- **URL of related IMPACT entry**: IMPACT (Intervention Management Planning and Coordination Tool) [2] is the standard CERN tool to manage interventions that require a partial or complete stop of beam operations within a specific accelerator. Among the various IMPACT entries, the coordinator of a TS or YETS will sequence, and schedule interventions required before resuming beam operations. While an infrequent scenario, there are instances when controls changes mandate the creation of an IMPACT entry. This is carried out to ensure a reserved testing time slot under specific conditions, which may include deactivating beam production. These controls changes predominantly concern a critical part of the machine, such as main magnet controls or RF cavity controls, as they could yield adverse consequences if unforeseen malfunctions arise. Consequently, such changes are visible to both SU and IMPACT coordinators.
- **URL of EDMS document:** EDMS (Engineering Data Management Service) [3] is CERN's standard repository for official documentation. Depending on the intricacy or impact of a controls change on operational interfaces, an EDMS Controls Change Request (CCR) document may be requested as part of the approval process. A CCR document follows a standard template to provide a detailed, formal technical description of a change and is subject to approval by pertinent stakeholders.
- **Integration with Accelerator Schedules**: the SU web application is seamlessly integrated with the Accelerator Schedules application, which is used to define and publish the annual schedules for the accelerator complex, alongside many other localized schedules. Both applications share common GUI components and their integration facilitates the presentation of the roster of controls changes that have been declared for a specific Technical Stop (TS) by simply clicking on the

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU2A004

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

corresponding component within the Accelerator Schedules application.

- **Automated email notifications**: an email is sent out on a weekly basis to the Operations teams containing a list of controls changes that have been declared in the past seven days. These changes are organized by accelerator to guarantee that the Operations team responsible for a specific accelerator only receives the corresponding changes. Another email is dispatched to the person who declared the controls change with a warning that the deadline for approving this change is approaching.

Technology

The SU application is one module of a multi-purpose web application called ASM (Accelerator Schedule Management) [4]. ASM is a web-based platform developed using the common technology stack used for CERN Accelerator Controls applications.

The backend stateless server is developed in Java, using the latest version of Spring (notably Spring Boot) framework.

The frontend, or presentation layer, is created with the latest version of the Angular framework from Google, and an in-house built library using Angular Material ensures that all web applications share the same look and feel.

The persistence is realised using an Oracle database. It is worth mentioning that no Oracle-specific features are required for ASM which means that ASM can be run on any relational database.

The common development and deployment process relies on Gradle, GitLab CI and Ansible in order to minimize the effort for managing releases and delivering new versions.

Monit is used to manage the redundant processes running behind HAProxy for high availability.

CURRENT STATUS

The experience gained during each TS or YETS has been systematically applied to significantly enhance the SU process, elevating it to a state of advanced maturity. Presently, it is regarded as a pivotal contributor to the performance and stability of the CERN Control system, demonstrated through swift and seamless resumptions of beam operations after each TS or YETS.

Furthermore, the SU process has developed a culture of best practices among the developers from the 11 groups providing controls solutions. The developers have become accustomed to adhering to meticulous preparation and validation procedures when preparing their controls changes. This includes measures such as code reviews, dependency checks, and formal requests for dedicated validation.

In the following two sections, recent advancements within the SU process are outlined. These are driven by the overarching objective of further enhancing the efficiency of deploying controls changes into the operational environment.

TOWARDS AN EXTENDED SU

Driver

In 2022, building upon the acknowledged success of the SU process and prompted by operational demands, the scope of the Controls Smooth Upgrades was expanded. This expansion entailed the incorporation of a more organized and formal approach to overseeing alterations in the software interfaces of control devices operating on Front-End (FE) computers. These interfaces play a crucial role in the seamless deployment process and integration within the Operations toolset. The new approach addresses two main objectives:

1. Ensure that FE software developers allocate adequate time in their planning, for the results of their work, to be integrated into higher-level operational software (this is typically done by someone other than the FE developer). Currently, new versions of FE software device classes can arrive too late in relation to the commissioning schedule. This in turn leaves insufficient time for the comprehensive integration and testing of the entire controls stack.
2. Establish an early approval process by the Operations team for new FE software versions. This is intended to guarantee that the operational interface and behaviour are optimized towards the needs of the Operations. At present, the interface and behaviour of a new FE software device class are not always discussed during the design stage with Operations. Consequently, the final interface might not align with the requirements of the Operations team, leading to multiple iterations needed for adapting GUI software, for instance.

Process

To address the aforementioned objectives, a new mini-project-based approach has been established to manage FE software evolution. This approach includes the following aspects:

- **Clear project-like planning:** defining stakeholders, deliverables, and milestones. Establishing collectively agreed upon milestones can help ensure the timely attainment of deliverables. For example a "FE software readiness" milestone helps clearly define when the subsequent integration within GUI components can start.
- **Data collection for dependency characterization:** gathering run-time data to assess whether and which dependent software components need to be adjusted. This primarily concerns applications like GUIs, data logging systems, and monitoring systems.
- **Project supervision:** overseeing the process is a project leader to orchestrate the activities, monitor the progress, and ensure overall technical coherence. This role aims to reinforce the communication between Operations and equipment-specific controls teams, when needed, based on the complexity of the systems involved and the changes foreseen.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU2A004

To pragmatically implement this strategy, the existing SU process and its underlying framework have been extended to incorporate new functionalities. The newly devised process has been named the "Extended SU", with the objective of accommodating these extended requirements.

New features

The SU web application has been enriched with additional attributes (as presented in Fig. 4) to comprehensively outline modifications made to FE software:

- List of dependent software requiring adaptation.
- Milestone indicating readiness of the FE software change for integration into the dependent software systems.
- Milestone for pre-validation: pre-validation procedures are conducted to ensure the functionality and compatibility of the software changes.
- Formal technical specification of the Controls Change Request (optional): if applicable, a formalized technical specification in the form of a Controls Change Request document can be included. This document serves to provide a detailed description of the proposed changes and their intended impact.

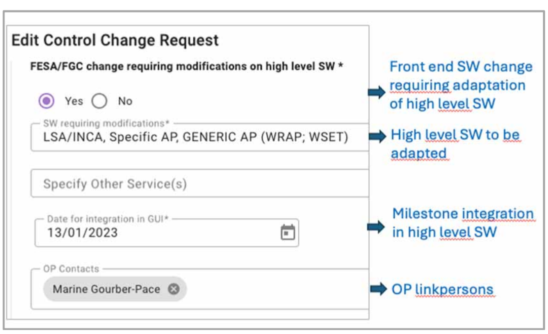

Figure 4: Additional features in SU web application.

Coordination Options

Although the primary objective of the Extended SU is to establish a more structured and formalized method for coordinating the integration of FE software into operational software, a deliberate effort has been made to ensure that this new process remains agile and adaptable. The aim is to prevent any unnecessary burdens on the collaborative efforts between FE software developers and the Operations team.

To achieve this, careful attention has been given to maintaining flexibility. The role of the Extended Smooth Upgrades coordinator is designed to be adjustable based on the specific needs of each mini-project. The level of coordination to be applied can range from a series of periodic progress checkpoints when collaboration between developers and Operations is efficient, to a more closely managed technical supervision involving regular meetings when needed.

The degree of coordination required is determined for each individual mini-project by both the Operations team and the initiator of the FE software change. This approach ensures that the level of involvement aligns with the specific demands and circumstances of each mini-project.

Status and Plans

The 'Extended SU' initiative was officially endorsed in September 2022. It was introduced as a proof of concept for the YETS occurring between 2022 and 2023. Following its implementation, a retrospective assessment was conducted in the spring of 2023. This review acknowledged the evident benefits of the initiative and recommended its continued utilization for future TS and YETS periods.

INCLUSION OF DEVICE MONITORING

Rationale

The SU process has proven to be effective and has helped establish a positive transformation in the mindset of controls experts when it comes to planning, coordinating and executing controls changes. Nevertheless, this process relies on a static declaration of upcoming changes and is therefore not completely immune to human mistakes. Typical scenarios are the declaration of changes where some dependencies have been unintentionally overlooked or when high-priority changes outside a TS or YETS are insufficiently validated.

The risk of such mistakes should not be underestimated, given the substantial volume and diverse origins of the controls changes intended to be declared within the SU process. These occurrences even if isolated can result in negative repercussions for the production environment.

To mitigate such risks, initiatives have been launched to further enhance the SU process and make the overall deployment of controls changes more robust. The most significant of these initiatives is focused on the management of changes to operational devices.

Terminology and Context

A "device" represents a piece of equipment within the CERN accelerator complex that can be controlled remotely. An "operational device" specifically pertains to equipment physically situated within an accelerator and utilized for beam operation. This stands in contrast to a "development" device, which is utilized for testing purposes in a laboratory setting.

The "device-property model" is integral to CERN's accelerator controls structure. It outlines the framework, along with the necessary operations and behaviours, for data exchange between low-level and high-level software components. Two fundamental concepts within this model are the "device" and the "property." Each equipment item (for instance, a power converter or a beam loss monitor) is considered a device. Meanwhile, a property denotes an attribute of the device – for instance, beam losses, threshold settings, or energy levels for a beam loss monitor. A device's property can be read and/or written.

This model adheres to a straightforward object-oriented approach, where objects are instances of classes. Consequently, devices themselves are instances of classes, and it is within these classes that properties are defined.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU2A004

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

Today, the Device Server Framework (DSF) defines a unified device-property model for CERN devices and provides a reference device server implementation.

The metadata describing the various device-property models are centrally managed within the Controls Configuration Service (CCS) [5] which includes a database, GUIs and APIs to enable users to maintain the control system configuration data in a self-service manner.

The Controls MiddleWare (CMW) system [6] provides the communication protocol and supporting services for exchanging data between devices and high-level software.

Device Monitoring Solution

To further enhance the SU process, monitoring has been established that identifies:

1. Modifications carried out on the device interface (e.g. a change in the devices property model)
2. Interconnected software components interacting with the device, that might require code adjustments to maintain compatibility with the new device interface.

Based on this information, two distinct notifications are triggered upon detection of operational device changes:

1. To the Operations team responsible for the accelerator where the device resides.
2. To individuals responsible for software applications, depending on the device.

Implementation for Notification 1

The breakdown of the steps involved in the notification process is as follows:

- DSF identifies when there's a modification in the device interface and initiates a call to the CCS API.
- The CCS engine reads from the database the accelerator(s) linked to the operational device(s) and the Operation linkpersons for the accelerator(s).
- The CCS engine notifies both the linkpersons and the responsible for the specific class version. The email notification includes a comprehensive description of the change, highlighting the variations compared to the previously deployed class version.

This process is in place since June 2023 and has already highlighted some incompatible changes deployed on operational devices.

Implementation for Notification 2

- For an operational device that has undergone changes, the CCS engine extracts a list of operational applications that have interacted with this device within a defined time frame (typically over the last 12 months, encompassing an entire beam run). This list of interactions is derived from log messages, captured via CMW and stored in an OpenSearch [7] system.
- The CCS engine queries the individuals responsible for the identified applications and subsequently notifies them about the modifications to the device. Furthermore, they are informed of the potential need to adapt their application's code to align with any device interface changes.

Challenges

The difficulty for the implementation for Notification 2. lies in accurately extracting the appropriate individuals responsible for the applications from the CMW-captured log message. Presently, there are instances where the information retrieved from the CMW log messages is misleading, as described in the following scenarios.

Many operational applications interact with device data via the centralised INCA [8] data acquisition server. As such, the telemetry data acquired from the CMW log file displays a general identifier linked to the INCA server, and not the actual user-facing operational application. To resolve this, an additional correlation step is needed to cross-reference the telemetry data from between the INCA server and application interactions, as well as from the CMW log message corresponding to communications between the INCA server and the underlying devices. This would aid in accurately identifying the responsible parties for the applications concerned.

Another constraint arises from applications relying on UCAP (Unified Controls Acquisition and Processing) [9] to access operational devices. At present, those applications do not have telemetry data captured via the CMW log messages, which prevents the retrieval of the individual responsible for the application. Addressing this issue requires further developments in the CMW and UCAP services to capture the required telemetry data .

The deployment in production of Notification 2. is planned for the upcoming YETS (November 2023 – February 2024) where a large number of code changes on operational devices is expected.

LESSONS LEARNT

The Smooth Upgrades success can be attributed to several key factors:

- A standardized process to declare, approve, deploy and validate controls changes.
- An efficient and integrated set of tools to support this process.
- A pivotal coordination role to ensure a proper communication among stakeholders all along the process, and essential cross-team coordination.
- A continuous effort to review and refine the process and supporting tools based on feedback gathered following TS and YETS events.

During the periodic SU reviews, the point to enlarge the scope to hardware upgrades has been discussed. Compared to software, the evolution cycle of controls hardware is less dynamic (hardware components are typically upgraded on a 3-5 year basis). The scope of dependent packages for hardware components is limited to the driver software, while each hardware replacement or upgrade is followed by a validation procedure with dedicated time allocated during the post-YETS commissioning phase. For these reasons, hardware changes are considered by Operations to be less critical than software ones in terms of risk of unexpected impact, and therefore they do not require to be managed by the SU process.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU2A004

CONCLUSION

The "Controls Smooth Upgrades" initiative has an ambitious goal: to offer Operations a comprehensive overview of all control modifications prior to their deployment during a TS or YETS. Additionally, it aims to guarantee that no changes are deployed outside of a TS or YETS, unless they are deemed critical and are specifically requested by the Operations team.

This paper emphasizes the importance of a systematic approach when implementing changes to accelerator controls during beam operation. The Smooth Upgrades procedure has proven to be a key contributor to the performance and stability of the CERN control system, by ensuring a seamless restart of the beam after a stop.

Today, the coordination of software interface changes for devices running on front-end computers is a critical aspect to be managed in order to achieve a seamless deployment process. This is the focal point where efforts are now concentrated, combining a comprehensive pre-deployment description of changes together with a monitoring of the changes that have been deployed.

REFERENCES

[1] M. Arruat *et al.*, "Front end Software Architecture", in *Proc. ICALEPCS'07*, Oak Ridge, TN, USA, Oct. 2007, paper WOPA04, pp. 310-312.

[2] C. Garino *et al.*, "Intervention Management from Operation to Shutdown", in *Proc. IPAC'13*, Shanghai, China, May 2013, paper THPWA035, pp. 3705-3707.

[3] `https://edms.cern.ch/ui/#!master/portal/tab?home`

[4] B. Urbaniec *et al.*,"Accelerator Schedule Management at CERN", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 580. doi:10.18429/JACoW-ICALEPCS2019-MOPHA149

[5] L. Burdzanowski *et al.*, "CERN Controls Configuration Service - a Challenge in Usability", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 159-165. doi:10.18429/JACoW-ICALEPCS2017-TUBPL01

[6] J. Lauener *et al.*, "How to design and implement a modern communication middleware based on ZeroMQ", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 45-51. doi:10.18429/JACoW-ICALEPCS2017-MOBPL05

[7] `https://opensearch.org/`

[8] S. Deghaye *et al.*, "CERN proton synchroton complex high level controls renovation", in *Proc. ICALEPCS'09*, Kobe, Japan, Oct. 2009, paper THA005, pp. 638-640.

[9] L. Cseppento *et al.*, "UCAP: a framework for accelerator controls data processing @ CERN", in *Proc. ICALEPCS'21*, Shanghai, China, Oct. 2021. doi:10.18429/JACoW-ICALEPCS2021-MOPV039

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI