

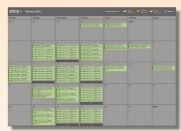
Lukasz Goralczyk†, Alexandros Foivos Kostopoulos, Brad Schofield, Jean-Charles Tournier

Scheduling Challenges

SCADA upgrades are a big campaign at CERN involving teams responsible for all types of industrial control systems.

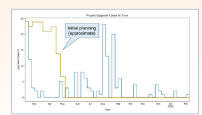


And many other...



Preparations started months before the actual beginning. A draft of the schedule has been created. Testing and validation has also started.

COVID-19 pandemic put upgrades on hold and later forced a complete reschedule.

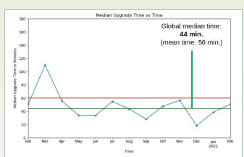


- Principles followed for the schedule:
- 1) Follow recommendations from control domain teams
 - 2) Group applications by their control domain
 - 3) Group applications by the server they are running on
 - 4) Avoid Mondays and Fridays
 - 5) Insert gaps to leave room for reschedule and other tasks

Scheduling changed to a more ad-hoc approach. Any interventions were agreed with application owners. Majority of upgrades were done remotely.

Good and Bad Experiences

New updated tooling proved to be easy to use and very reliable. Median time of upgrade was decreasing as more experience was gained and problems fixed.



Despite major efforts in automation manual steps still existed which slowed down the whole process.

Upgrade Step	Manual	Time in Minutes
Update of application startup scripts	Yes	6
Pre-upgrade scripts	Yes	3.5
Post-upgrade backup	Yes	3.5
JCOP and UNICOS components upgrade	No	2.5

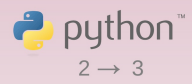
Long Shutdown 2 period provided a less strict schedule limits allowing other activities not related to the upgrade.

Testing and validation did not prevent all bugs, but thanks to less strict schedule there was time to fix them.



Tooling

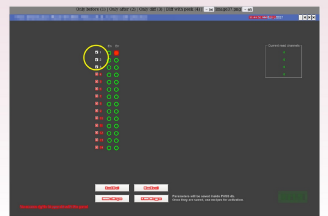
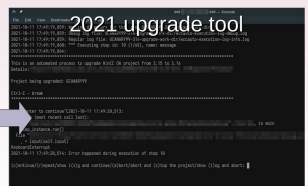
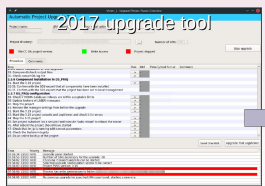
A new Python interface for WinCC OA has been developed using Python 3. It was partially based on a previous library which used Python 2. It's a crucial part of other tools.



- Some of the improvements include:
- Consistent and richer API
 - Understandable error messages
 - Reliable and faster

New tools have command line interface (CLI), a different approach from previous upgrade which used a graphical user interface (GUI). CLI requires only a secure shell connection, automates better and it's easier to document.

Other tools have been developed to aid testing and upgrade verification. Here synoptic view comparison tool. Differences are highlighted with red.



Future Improvements



Shorten upgrade time by automating remaining parts of the process.

Develop a tool to aid schedule creation and modifications, given various constraints.



Offline upgrades, which, this time, were only used for testing and validation.

Use Ansible instead of in-house developed tool for step processing.

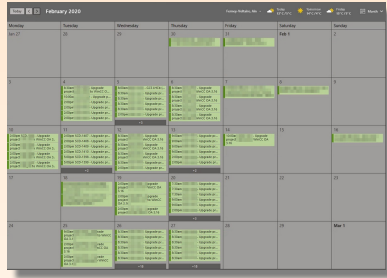


Scheduling Challenges

SCADA upgrades are a big campaign at CERN involving teams responsible for all types of industrial control systems.



And many other...

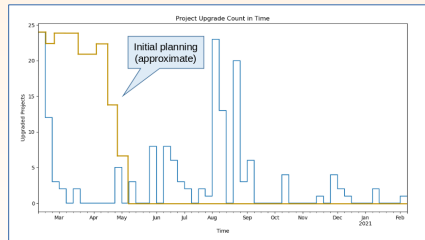


Preparations started months before the actual beginning. A draft of the schedule has been created. Testing and validation has also started.

Principles followed for the schedule:

- 1) Follow recommendations from control domain teams
- 2) Group applications by their control domain
- 3) Group applications by the server they are running on
- 4) Avoid Mondays and Fridays
- 5) Insert gaps to leave room for reschedule and other tasks

COVID-19 pandemic put upgrades on hold and later forced a complete reschedule.

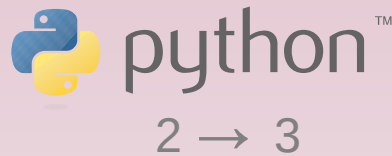


Scheduling changed to a more ad-hoc approach. Any interventions were agreed with application owners.

Majority of upgrades were done remotely.

Tooling

A new Python interface for WinCC OA has been developed using Python 3. It was partially based on a previous library which used Python 2. It's a crucial part of other tools.

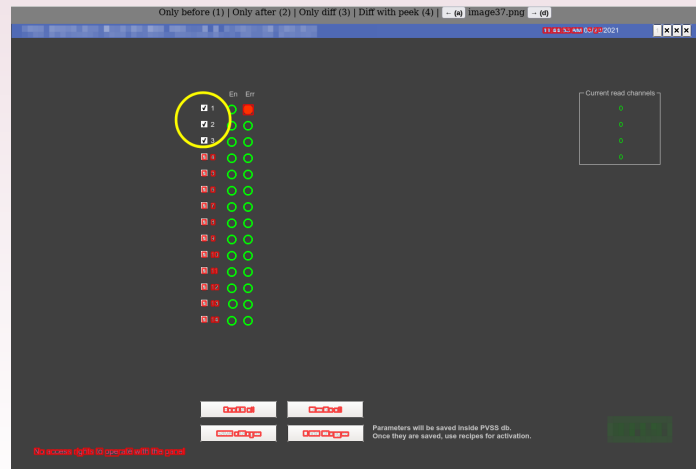
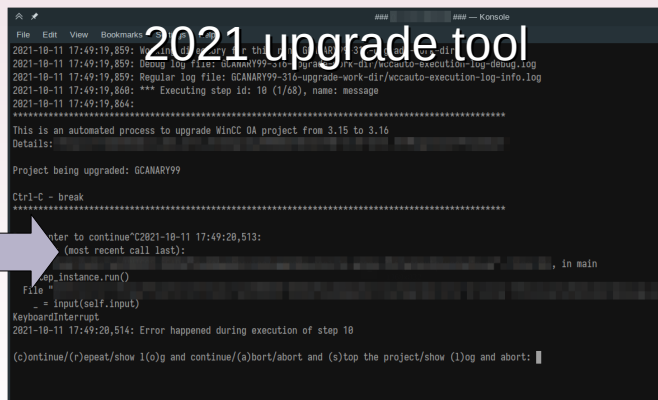
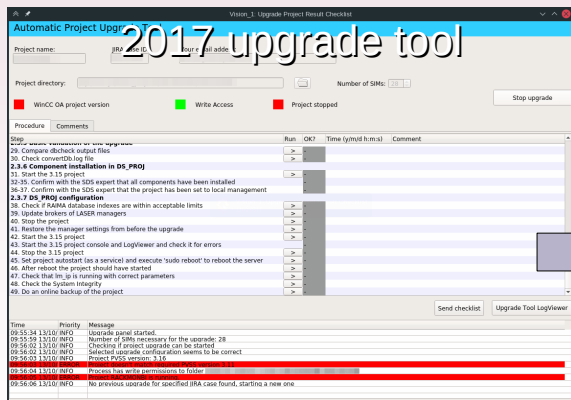


New tools have command line interface (CLI), a different approach from previous upgrade which used a graphical user interface (GUI). CLI requires only a secure shell connection, automates better and it's easier to document.

Some of the improvements include:

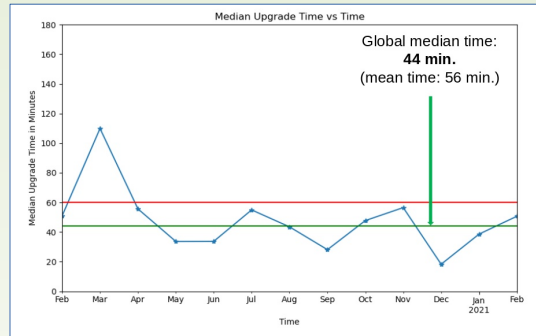
- Consistent and richer API
- Understandable error messages
- Reliable and faster

Other tools have been developed to aid testing and upgrade verification. Here synoptic view comparison tool. Differences are highlighted with red.



Good and Bad Experiences

New updated tooling proved to be easy to use and very reliable. Median time of upgrade was decreasing as more experience was gained and problems fixed.



Long Shutdown 2 period provided a less strict schedule limits allowing other activities not related to the upgrade.

Despite major efforts in automation manual steps still existed which slowed down the whole process.

Upgrade Step	Manual	Time in Minutes
Update of application startup scripts	Yes	6
Pre-upgrade scripts	Yes	3.5
Post-upgrade backup	Yes	3.5
JCOP and UNICOS components upgrade	No	2.5

Testing and validation did not prevent all bugs, but thanks to less strict schedule there was time to fix them.

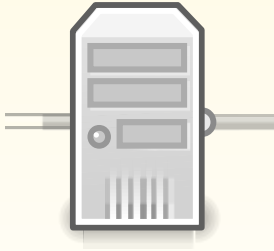


Future Improvements



Shorten upgrade time by automating remaining parts of the process.

Develop a tool to aid schedule creation and modifications, given various constraints.



Offline upgrades, which, this time, were only used for testing and validation.

Use Ansible instead of in-house developed tool for step processing.



ANSIBLE