SURPRISE! IT'S SPRING!

In high-technology environments, normal and expected events can bring surprises. In this article, **Chad Todd** gives an example of how something that is 'normally' a non-event can become an operational surprise: the US springtime change.

I was formerly employed by a healthcare IT vendor and worked in a department where approximately 300 customer systems (both production and non-production environments), were remotely hosted in data centres. These customer systems were managed by over 150 employees, with two different kinds of back-end operating systems, in various US time zones.

Each year these systems go through the fall (autumn) time change and the spring-time change. For both events, an 'all hands-on deck' approach is taken, and the employees come into the office around 22:00 hrs. Employees are divided into large rooms with one person designated as the leader of each room. The leader of each room has two main responsibilities: 1) to communicate the time zone to focus on, and 2) to dial into an open conference call bridge line. One of these large rooms is for subject matter experts (like me) to help troubleshoot possible system problems that other employees might experience during or after the time change.

For the fall time change where the time moves back one hour (02:00 to 01:00), systems are taken down by an employee for one hour until it is the 'new' 02:00 for the new time zone. The systems are then brought back online. There are many reasons for this, but one of the main reasons is to ensure the appropriate timestamp, and not duplicate pharmacy orders and medication (a patient safety issue). For the springtime change, the systems move forward one hour (02:00 to 03:00) according to the specific time zone. The systems are left up and running since timestamp duplication for pharmacy orders and medication is not a concern.

The 'Not' So Normal Springtime Change

Since springtime changes were not problematic in previous years, management and customers expected that each springtime change would be a success. What happened next revealed surprising problems.

When the eastern time zone systems (production or non-production) flipped from 02:00 to 03:00, an employee in another room reported something unusual to the chief room leader. During the post time-change systems check, the employee noticed the third-party software processes for one particular operating system were using all the system's central processing unit (CPU). Multiple rooms started reporting the same system behaviour across eastern time zone systems.

The troubleshooting room was alerted to this surprising system behaviour via the open conference call bridge line. Having gained expertise with this third-party software, I started to look at a non-production environment and confirm what was being reported from the various rooms. I took multiple actions to try to understand the internal system processes, using various operating system commands. However, with time ticking and springtime change for the central time zone coming closer, I decided to 'cycle' these third-party software processes (in the non-production environment). This meant

"No matter how technological the field, it is the humans that use adaptive strategies to handle surprises in these complex and dynamic environments to keep the systems running."

taking the system down entirely, as is done for fall-time changes. After performing the actions required to take the system down and then bring the system back up, third-party software processes were no longer using all the available CPU on the system. These same actions were taken in other non-production environments, which confirmed that cycling the system addressed the problem.

Following this, there was much coordination and communication between staff and customers, both in person (subject matter experts from the troubleshooting room walked around) and via digital means (chat and calls) about the need to cycle or take down the system.

After the event, we worked with the third-party vendor and replicated the problem. This helped to address the failure modes through a code patch provided by the third-party vendor.

Key Takeaways from the 'Not So Normal' Springtime Change

So why is this interesting? As is often the case with unwanted events, there are learnings that are both technical and social. The 'not so normal' springtime change is no different but what is interesting is that all this happened during a global planned systems maintenance event.

- **Widen the scope of validation activities:** It can be easy to miss 'normal but not everyday' events in the scope of validation activities. Expected events can create surprises. After the event it was recommended that fall and springtime changes be incorporated in the validation phase of new technologies or new system patches. This doesn't guarantee a future successful time change but reduces the risk of unwanted surprises.

- **The troubleshooting-takedown trade-off:** One of the most glaring dynamic trade-offs was deciding whether to keep troubleshooting further to capture more data or take down the system and bring it back up. The goal conflict arose because an answer was needed on how to mitigate the failure modes, since springtime change was going to occur for the other time zones that had not yet changed.
- **Prepare for adaptive choreography:** We achieved what is defined as adaptive choreography (Maguire, 2020), where we coped with an operational surprise along with dynamic demands for coordination across a large group of employees spread across multiple rooms. This requires planning to ensure appropriate communication channels and protocols.
- **Don't assume past success guarantees future success:** Past successes with the springtime change gave confidence to management and employees, essentially becoming the mindset that any time change is guaranteed to be successful. However, the key is to not take past success as a guarantee for future success (Dekker et al., 2008). It was the employee's adaptive capacity during the turbulence of the springtime change that led to the success of it.

This is just one story about an operational surprise in software operations. Such events happen all around the world. No matter how technological the field, it is the humans that use adaptive strategies to handle surprises in these complex and dynamic environments to keep the systems running.

Chad Todd has worked in software engineering, systems engineering and operations for over twenty years. Chad has particular interest in systems thinking, incident management, incident response, incident analysis, and resilience engineering. Chad holds an MBA and recently graduated with an MSc in Human Factors and Systems Safety from Lund University.

Reference

Dekker, S.W., Hollnagel, E., Woods, D.D., & Cook, R.I. (2008). Resilience engineering: *New directions for measuring and maintaining safety in complex systems Final Report*, November 2008.

Maguire, L.M.D. (2020). *Controlling the costs of coordination in large-scale distributed software systems* [Doctoral dissertation, Ohio State University]. OhioLINK Electronic Theses and Dissertations Center. https://bit.ly/3GYQjMK