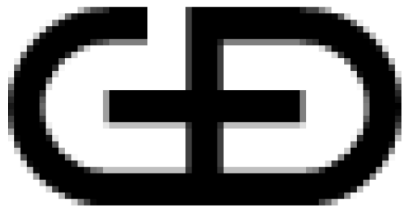


9th
UCAAT *User Conference on
Advanced Automated Testing*

**Increase Robustness
with Time Travel
Is It Possible and
Reasonable?**

Rosalinde Schuster Victor Navratil Stephan Schulz



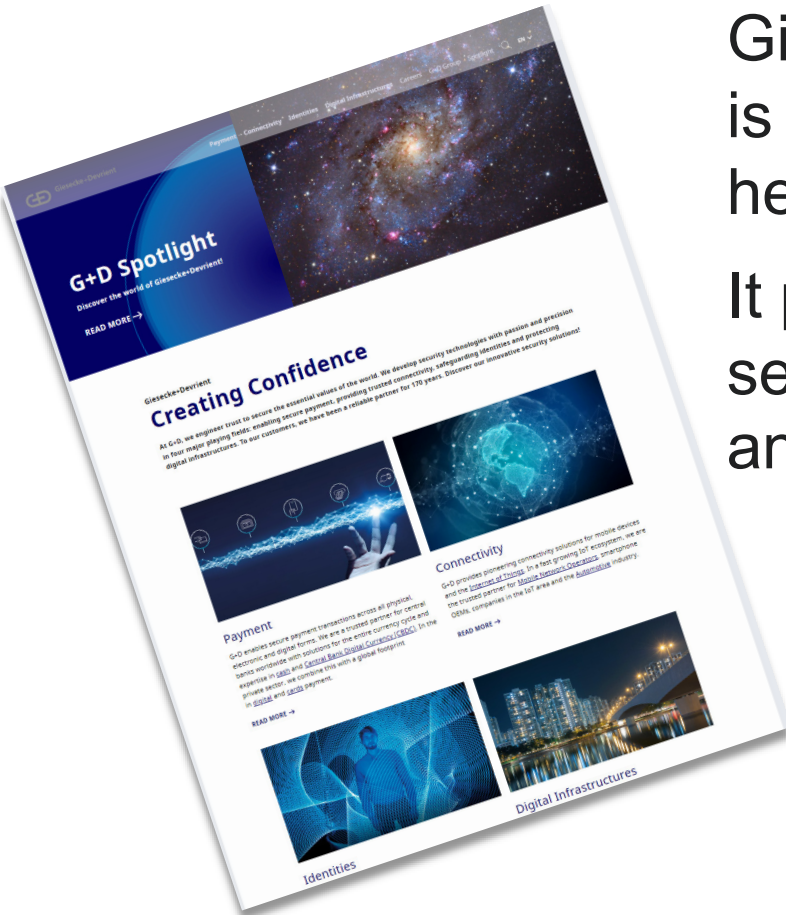
14.09.2022

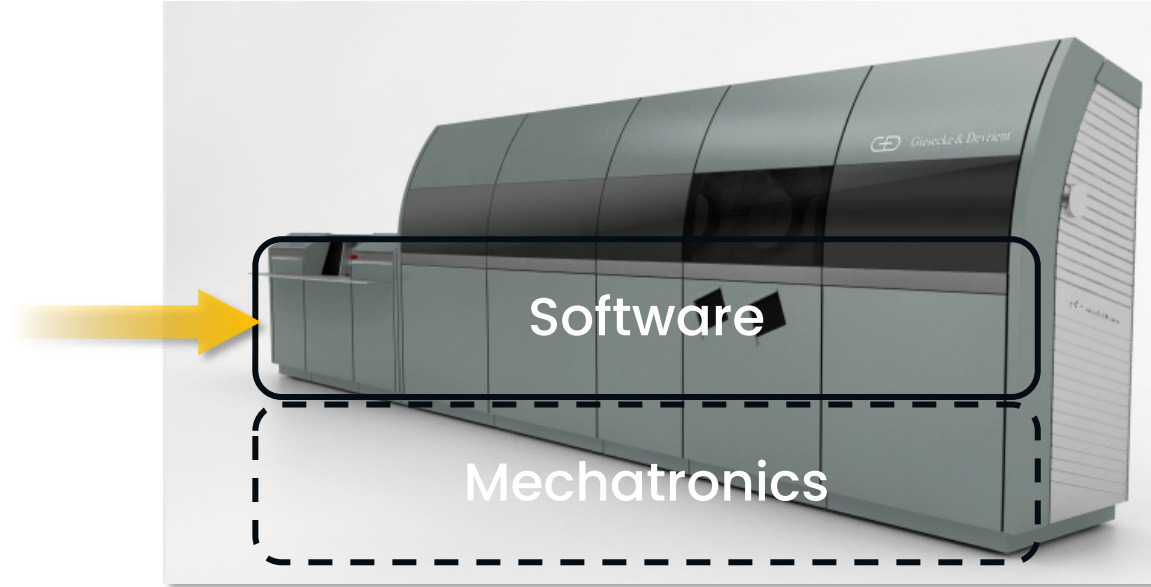


G+D Currency Technology

Giesecke+Devrient is a German company with headquarters in Munich.

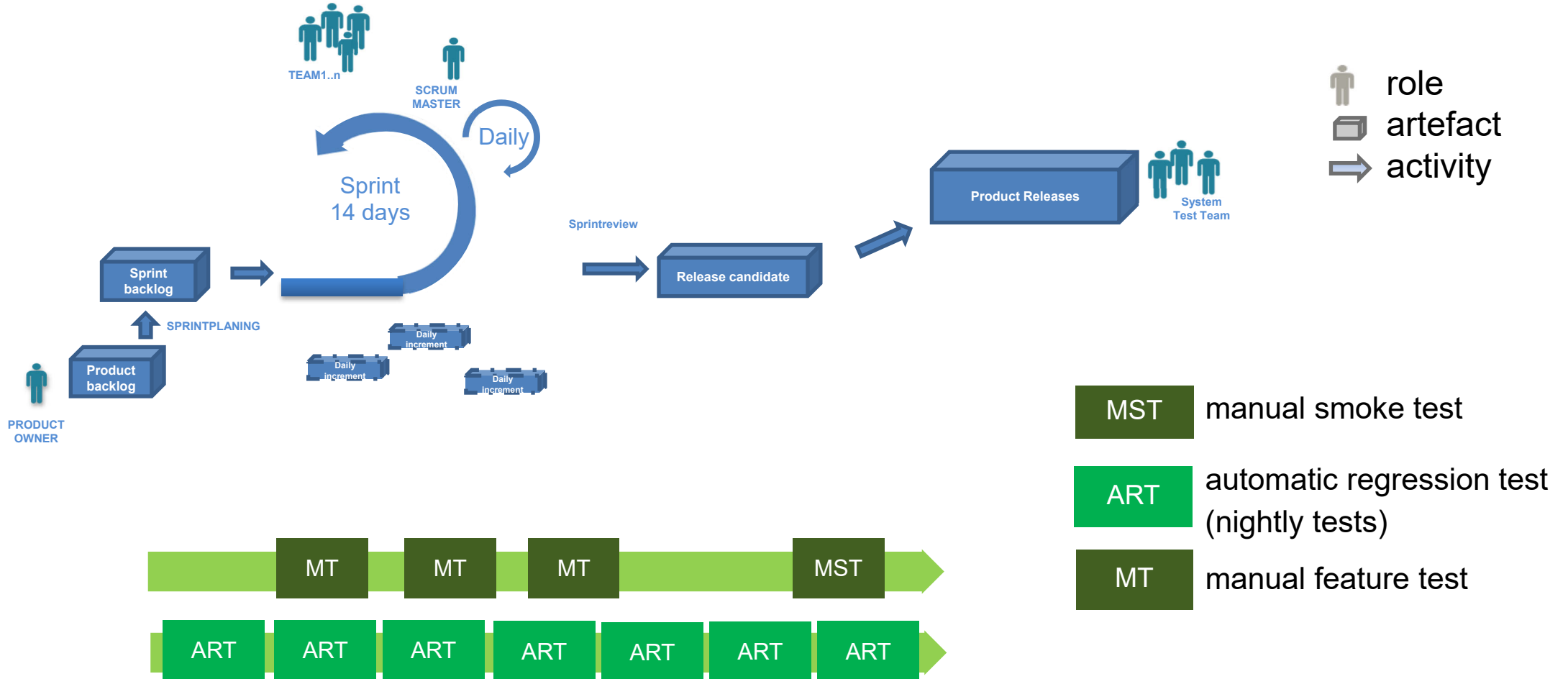
It provides banknote, securities printing, smart cards and cash handling systems.



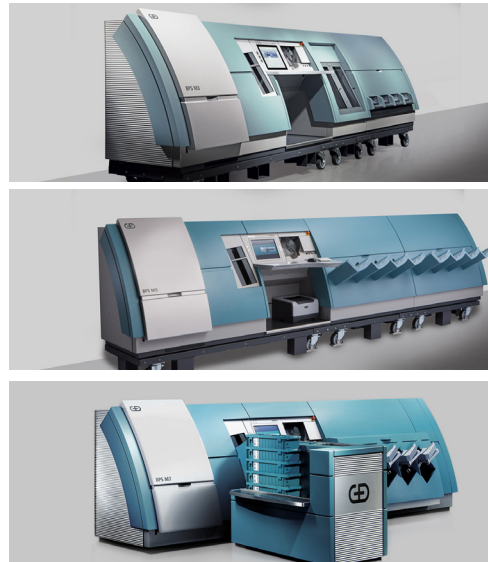
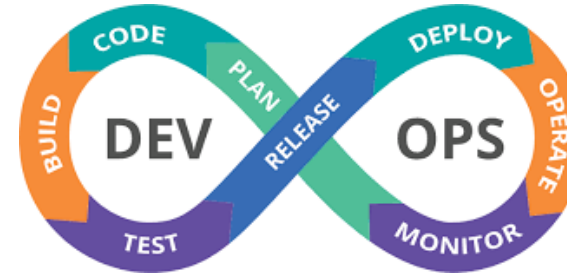
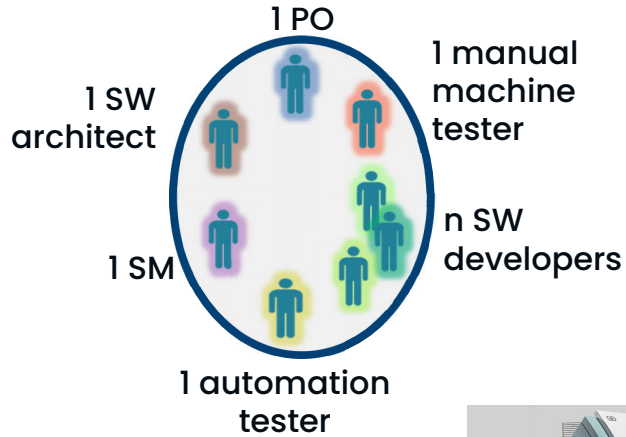


- Multiple products based on common product platform that are deployed worldwide in banknote printing, central banks, cash-in-transit centers, casinos, etc.
- Each processes millions of banknotes/day, 24/7 & is configurable for any currency

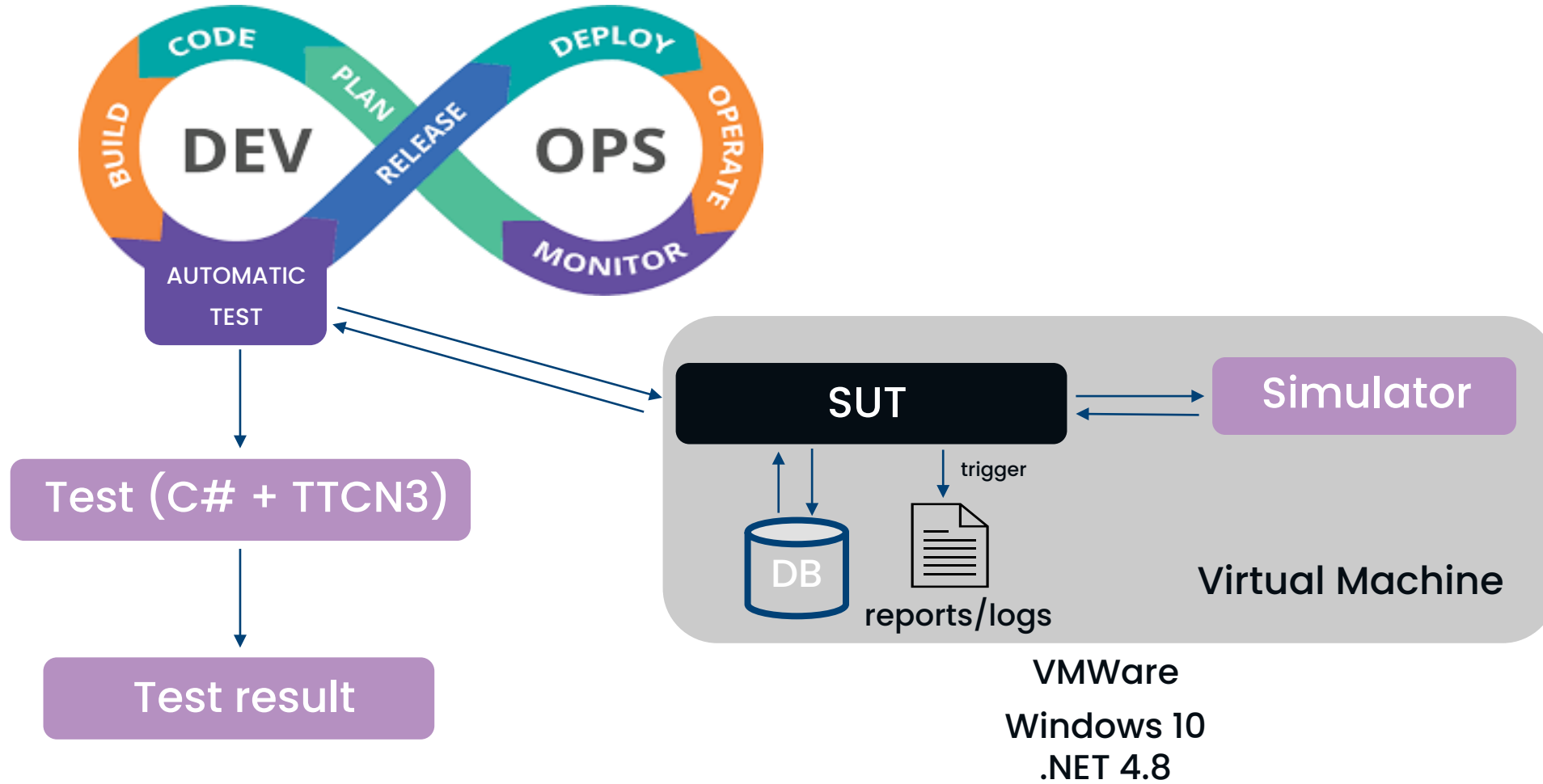
How we work – Scrum & CI-CD



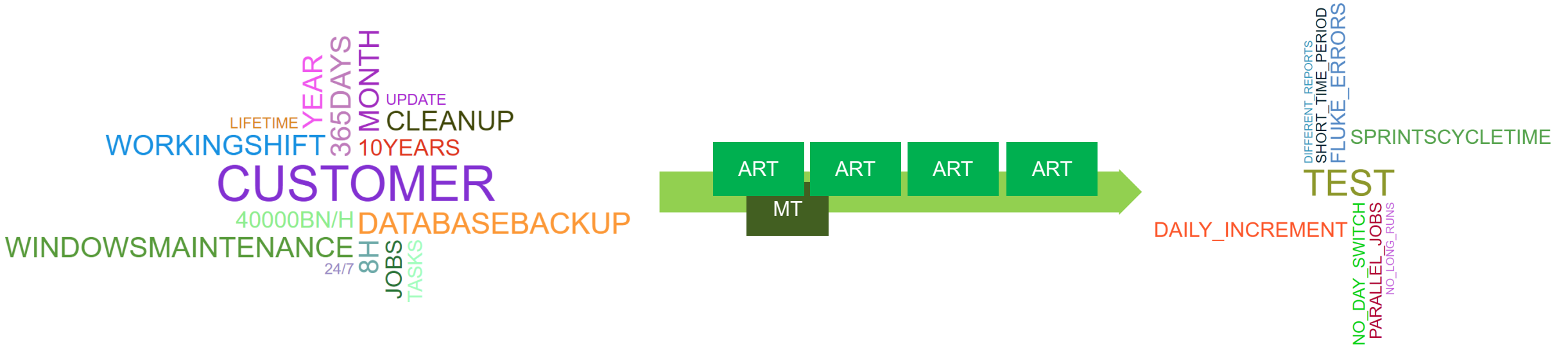
Our Scrum process setup



automatic regression test

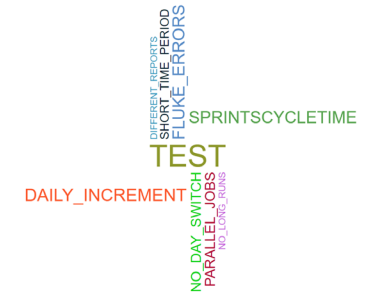


Motivation for testing time boundaries



We have to improve our nightlies!

Our problem: Very short time cycles



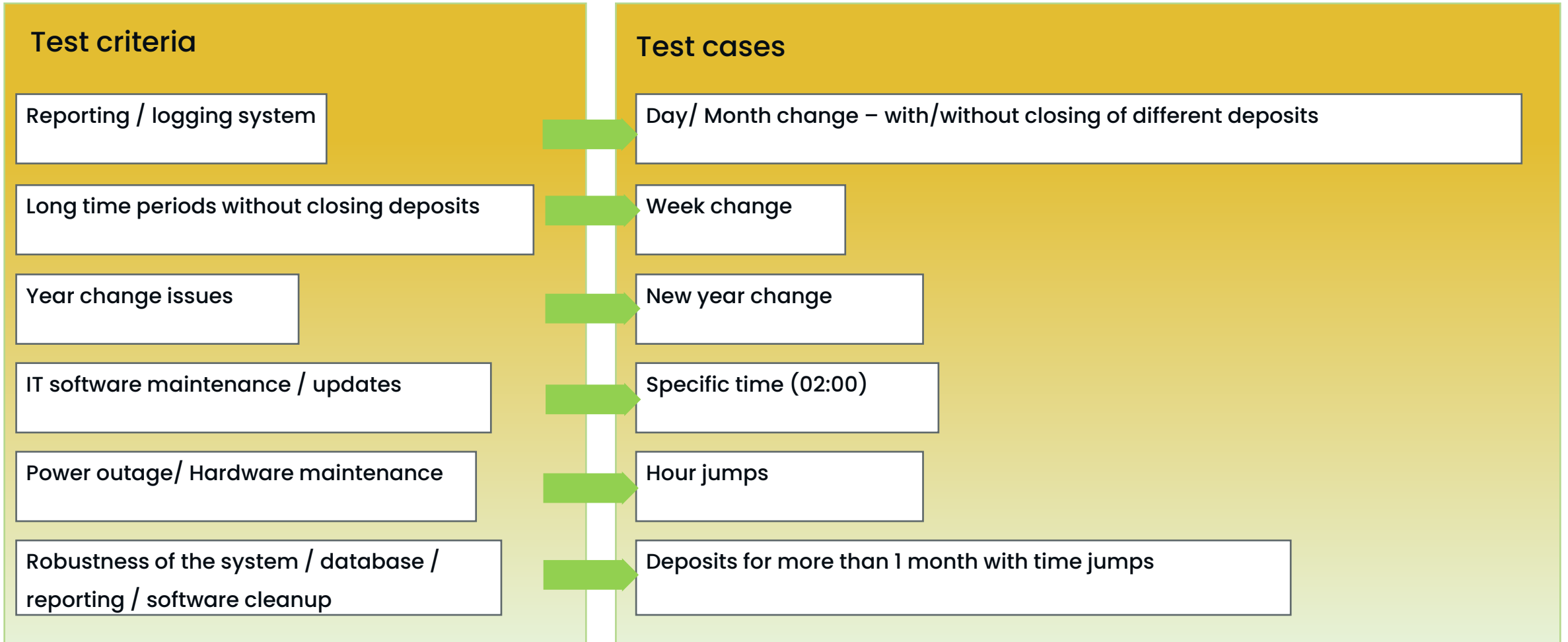
➔ No time for testing with real time

➔ need to manipulate day and time in tests



➔ better analysis possible because of repeatability

The hard part: Test cases



Easy part: the implementation

```
log ("[INFO] this instance is not necessary so applikation can be killed");
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c taskkill /IM Application.EXE /T /F"); // simulate crashed situation

f_delayMtc(5.0); // wait a little bit
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c powershell Stop-Service -Name ""Service1"" -Force "); // stop service
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c powershell Stop-Service -Name ""Service2"" -Force "); // stop service
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c powershell Stop-Service -Name ""Service3"" -Force "); // stop service

fx_vmRunCmd("PC", ""C:\Program Files (x86)\VMware\VMware VIX\vmrun.exe"" -I ws -gu UserName -gp UserPassword CopyFileFromHostToGuest E:\Image.vmx E:\SourceDevOpAgent\DeleteDatabase.cmd C:\DestinationOnVM\DeleteDatabase.cmd");
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c C:\DestinationOnVM\DeleteDatabase.cmd "); // Delete Database because too long period in future
log ("[INFO] Delete other artefacts too to simulate 'clean system'....");
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c del /Q C:\GeneratedReports\*.");
f_delayMtc(10.0); // wait a little bit because of database script
```

Clean Up tasks

Manipulate time

Startup tasks

```
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c powershell set-date (get-date ""*givenDate*""T*givenTime*""*.000000+01:00""); //future date SMS crashed
```

```
// maybe copy some further script for execution
// fx_vmRunCmd("PC", ""C:\Program Files (x86)\VMware\VMware VIX\vmrun.exe"" -I ws -gu UserName -gp UserPassword CopyFileFromHostToGuest E:\Image.vmx E:\SourceDevOpAgent\FurtherScript.cmd C:\DestinationOnVM\FurtherScript.cmd");
// fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c C:\DestinationOnVM\FurtherScript.cmd "); // Do something else
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c powershell Start-Service -Name ""Service3"" "); // restart service
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c powershell Start-Service -Name ""Service2"" "); // restart service
fx_vmRunCmd("CountingMachine", "C:\Windows\System32\cmd.exe /c powershell Start-Service -Name ""Service1"" "); // restart service
```

Do the main test

```
//simulate first start of counting machine with manipulated date
f_CountingMachine_StartAndConnect();
f_CountingMachine_logon("UserName", "Password");
....
```

Test automation is done by C# / TTCN3

We use powershell commands to manipulate date and time

Even more test case ideas

Test cases

Update of local hour (E.g.: after BIOS reset)

Change of time zone

Daylight saving time changes

2038-01-19 03:14:02 UTC (Unix systems)

Travel back in time (Motherboard battery failure)

...

Conclusions

- Initially our nightlies did not catch our problem with time boundaries
- Hard part turned to be the design of test cases
- Implementation turned out to be relatively simple
- We believe it is both reasonable and possible to automate such checks
- Today our nightly tests (& weekly test) check these time boundaries and lead to better robustness!



Any further questions?

Victor.Navratil@gi-de.com
Stephan.Schulz@gi-de.com
Rosalinde.Schuster@gi-de.com

