
Microsoft
Your performance testing approach
FastTrack for Dynamics 365 for Finance and Operations
Christopher LIM / Davy VLIEGEN
Taipei, Taiwan

About this Tech Talk

Tech Talk: Your performance testing approach

Objectives of this Tech Talk

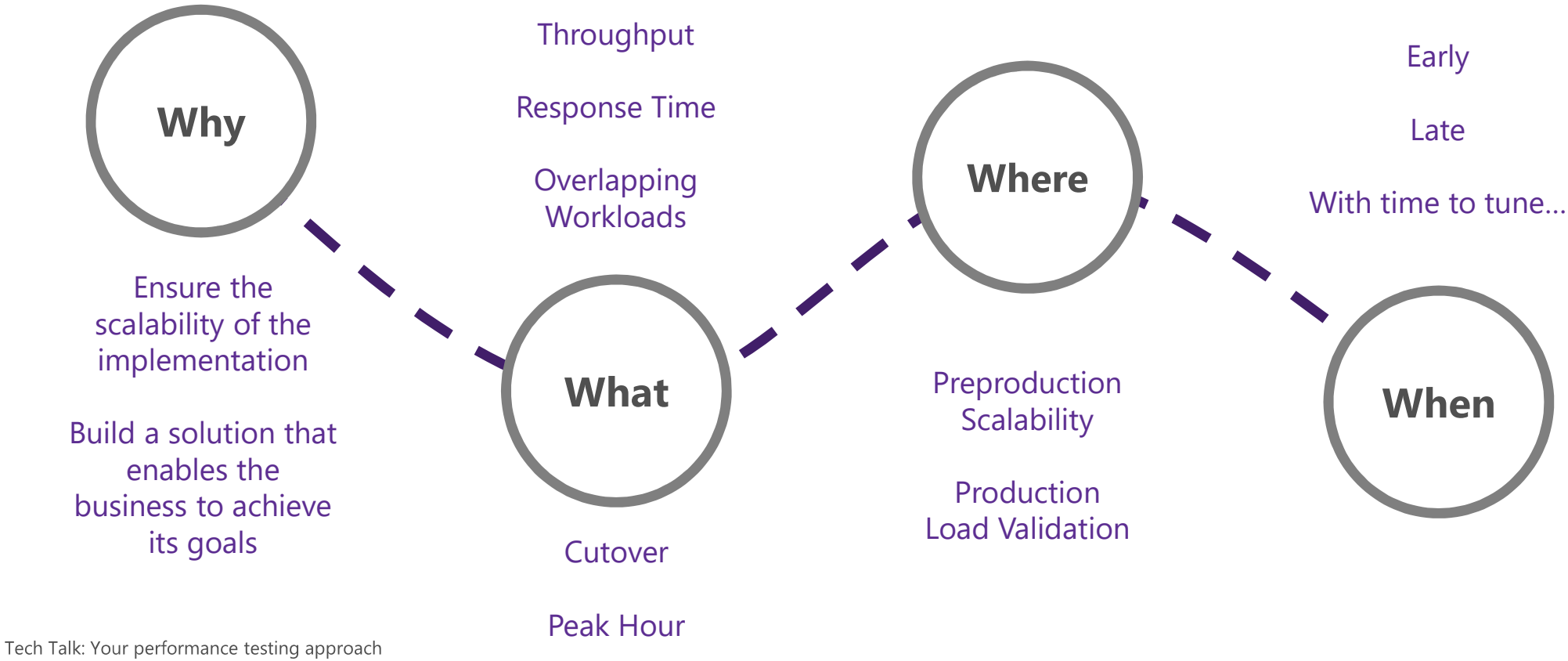
1. Help you **avoid** or **limit risks** of facing **performance issues** post Go-Live
2. Help you build an **adapted approach** to **performance testing**
3. Provide you **guidance** based on **experiences** from actual **implementations**

What it doesn't cover

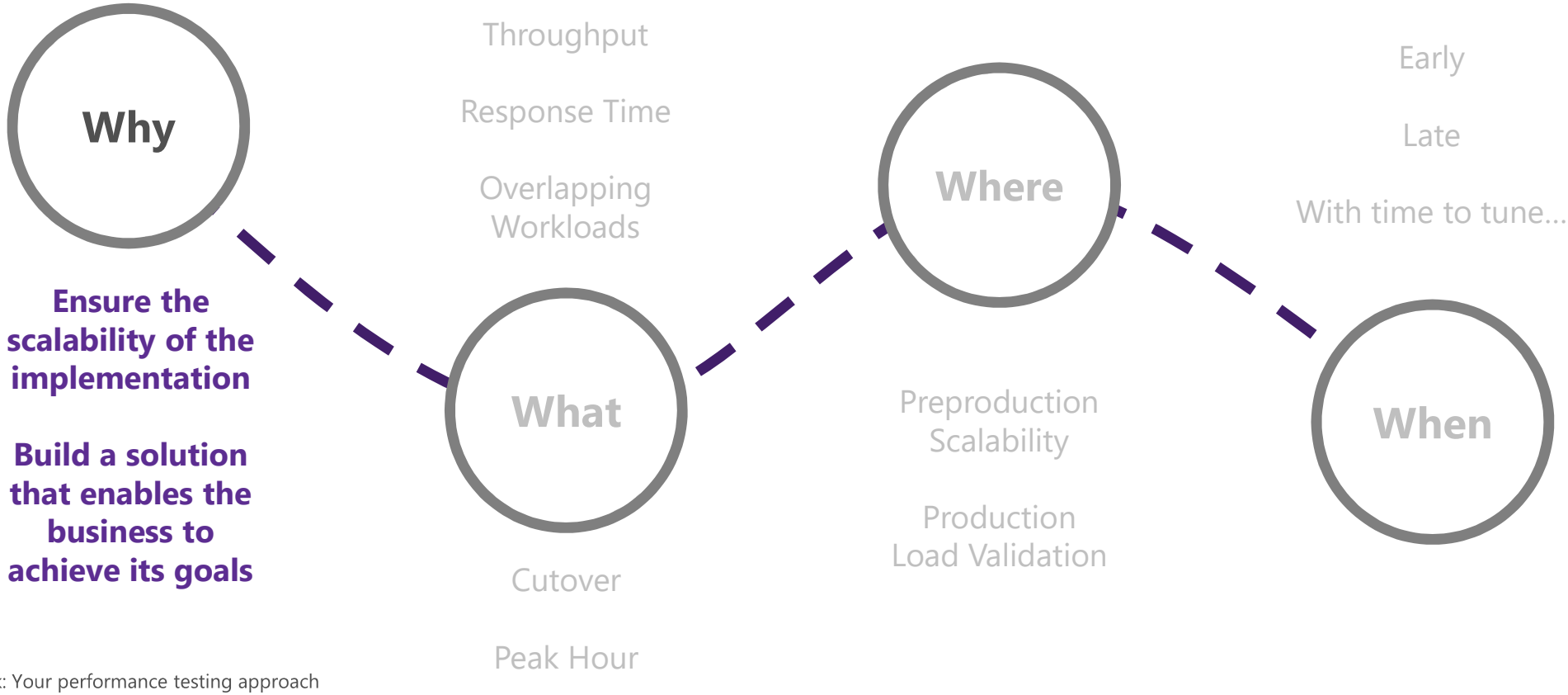
- In-depth presentation of the stress testing methdology
- In-depth demonstration of the tools
- Automated testing
- Performance best practices for development
- For the above subjects, look at the resources at the end of the presentation

Performance testing in a nutshell

Performance enhancing practices



Performance enhancing practices



Why would you do performance testing?

Think of…

1. A car that wouldn't be tested at maximum speed
2. An elevator that wouldn't be tested for maximum load
3. An e-commerce website that wouldn't be tested for maximum number of visitors
4. …

Why?

Performance testing ensures the scalability of the implemented solution

Additionally:

- It helps identify and investigate the performance issues early in the project
- It helps Identify optimization points to deliver the optimal value to the business
- It gives the assurance that the solution will enable the customer to achieve its business goals
- It can simulate projected volumes (e.g. in 2-3 years)

Why?

What parts of the solution can impact performance?

Processes and operations	Users	Analytics and reporting	Integration	Customizati-ons	Data
Servers	Database	Azure infrastructu-re	Standard solution	Device	Independant Vendor Software (ISV)

Why?

Microsoft's commitment is to build a scalable platform

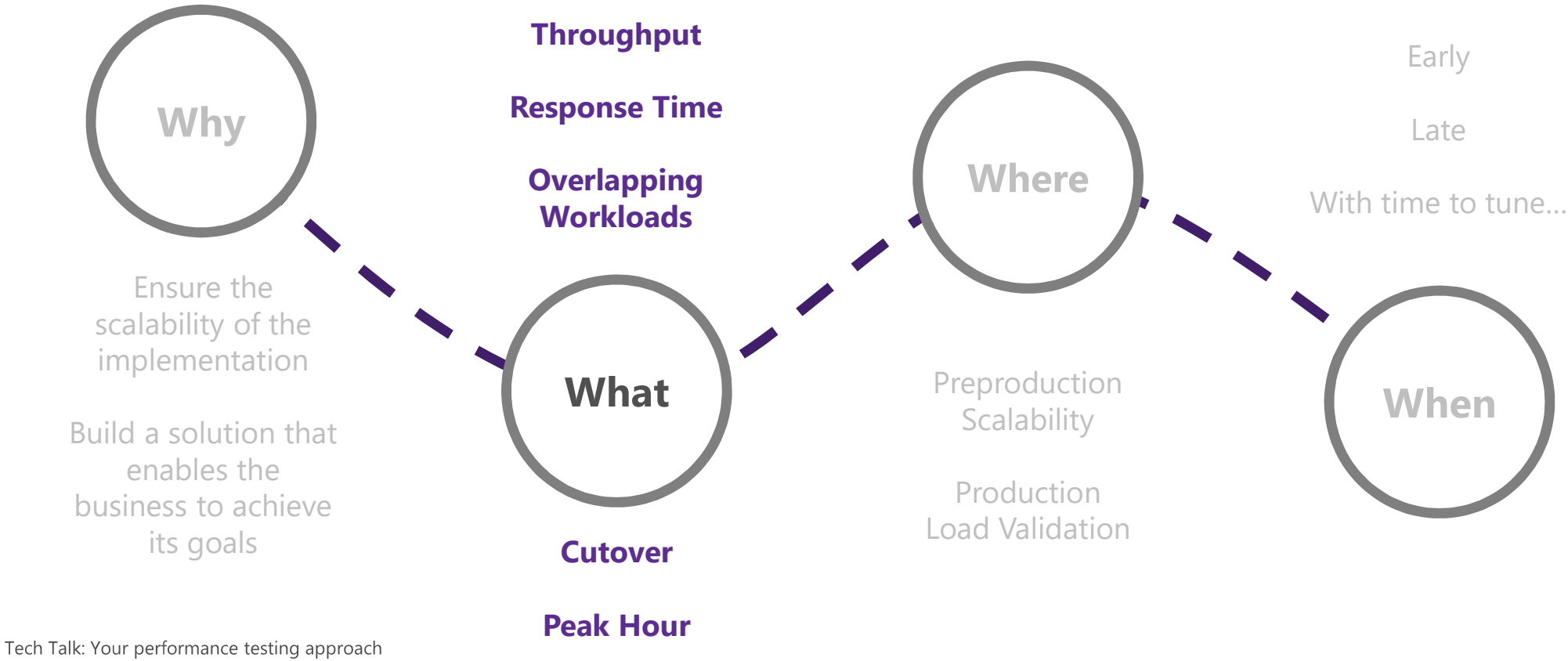
Processes and operations	Users	Analytics and reporting	Integration	Customizati-ons	Data
Servers	Database	Azure infrastructu-re	Standard solution	Device	Independant Vendor Software (ISV)

Why?

The parts built during the implementation can disrupt the scalability

Processes and operations	Users	Analytics and reporting	Integration	Customizati-ons	Data
Servers	Database	Azure infrastructu-re	Standard solution	Device	Independant Vendor Software (ISV)

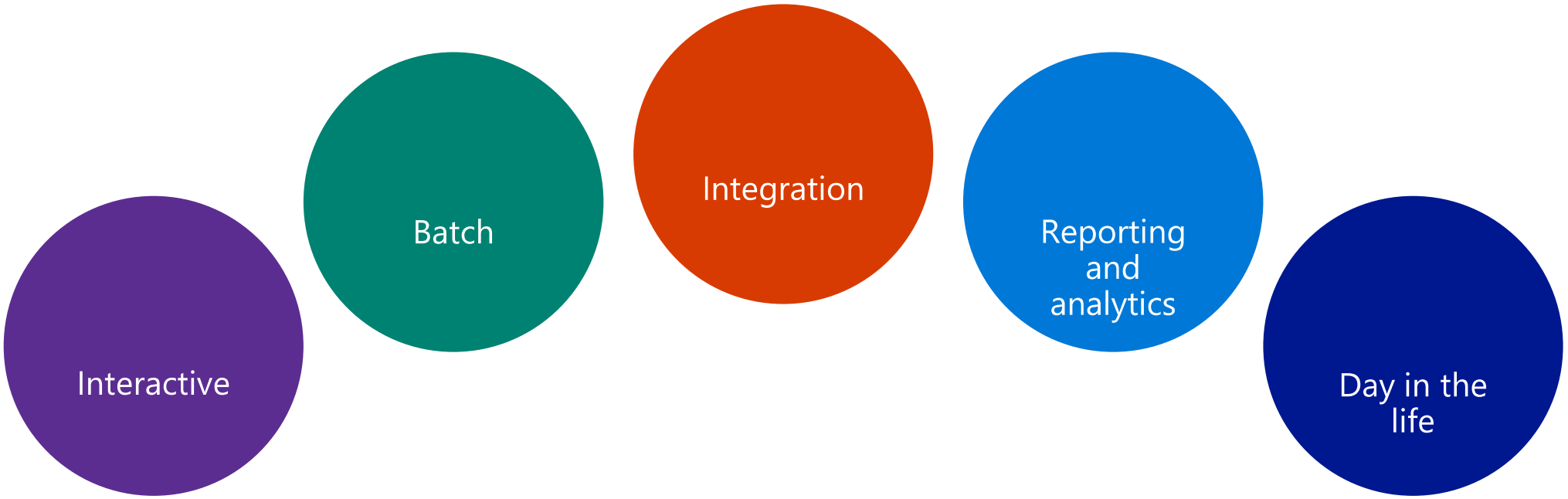
Performance enhancing practices



What performance test scenarios to cover?

From the business objectives to the critical scenarios

- Start with high-level business objectives (e.g. target number of customers, target number of sales, target number of stores, …)
- Define the expected normal and peak volumes for the functional areas in-scope (usage profile)
- Define the performance testing scenarios (key user-activities, batch processes, integrations and reports)
- For each scenario, define measured goals and constraints for response time and throughput (with reasonable expectations)
- Define your day-in-the-life, i.e. realistic workloads that would happen concurrently post Go-Live to help you model and simulate in the most realistic manner (especially at peak-hour)
- Define the preparation activities for each scenario (e.g. existing master and historic data in the system or creation of the file to import)



The common types of scenarios

INTERACTIVE SCENARIOS

Tasks/activities performed by the user through the User Interface

Specific critical interactive flows

Is the critical flow fast enough for the end-user to achieve its goal?
e.g. the creation of sales orders, verifying the stock of an item, …

Interactive activity that processes a large number of lines

How much time does it take to process all lines interactively? Is the activity going to freeze?
e.g. the payment proposal, the creation of very large purchase orders, …

The customized workspaces with additional tiles

Is my customized workspace loading?
In this case, each tile is linked to a query that can be more or less efficient

BATCH SCENARIOS

Tasks/activities executed in batch mode for bulk and parallel processing

Scalability of a batch job (10 records, 100 records, 1000 records, ...)

Does the duration/throughput remain constant or does it drop significantly?
e.g. the cart management

Massive volume batch jobs

The use of hardware can be compared by running the process on a Tier-2 and Tier-4/5 environment
As the number of threads available increases, does the performance increase as well?

Deadlocks identification

Within the batch job, do the parallel tasks generate deadlock or other issues?

INTEGRATION SCENARIOS

Exchanges/communication with external applications through the interfaces

Maximum load for the messages (whenever possible)

The interface works for a message with minimal lines, can it handle the forecasted maximum size?

Real-time interfaces

Is the response time acceptable for this real-time interface?
i.e. the source application wait for the target application to respond before continuing the process

Comparison of OData VS DMF

As OData isn't natively made to handle large payloads, is the interface more efficient with DMF?

REPORTING AND ANALYTICS SCENARIOS

Reports generation/consultation and analytics dashboards/visualizations

Reports generation, sending, printing and visualization

Are the reports being generated/loaded with an appropriate response time?
Are there issues when sending the report via email (e.g. message size limit)?

Approach validation (Entity store / BYOD / Odata)

Can the Power BI report be more efficient?

Power BI reports

Are my Power BI reports loading? Are they crashing?
e.g. if the reports use OData connectors to access D365FO, they may face performance issues

DAY IN THE LIFE SCENARIO

Simulating tasks/activities as they would happen during working hours post Go-Live

Activities like it would happen post Go-Live in PRODUCTION

One task tested uniquely works, what if all tasks are initiated at the same time?

i.e. test the interactive, batch, integration and reporting scenarios at the same time like it would happen during the peak hour

What?

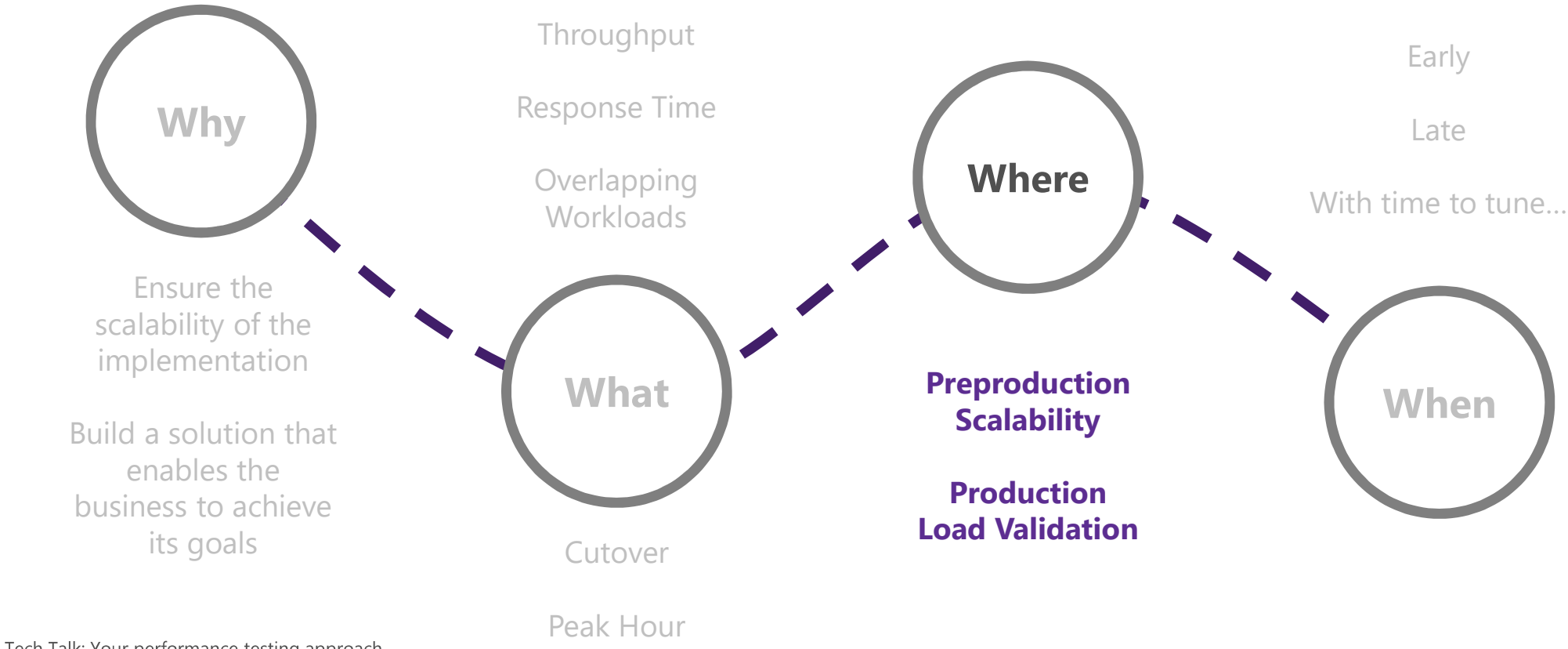
DON'T underestimate the preparation activities

- E.g preparation of the data through DMF or jobs

DO consider both the ideal target and maximum acceptable target response time / time window

- Finetuning a scenario should be done within specified boundaries
- Otherwise, finetuning could be performed endlessly

Performance enhancing practices



Where?

In essence, the **Tiers** differ in the number of available **threads**

- Put simply, an environment is a combination of compute (AOS) and database processing (Azure SQL)
- Threads handle tasks in the system (interactive, integration, batch, day in the life)
- The environment will provide a maximum number of threads depending on its computing and database processing capacities
- The higher the tier, the higher the number of threads
- An environment not adapted for your day-in-the-life scenario would be maxed out fairly quickly

Rely on your **partner** to select the environment Tier for performance testing

- Your partner will propose and recommend you the most adapted environment Tier given the business operations and objectives (Tier-2, Tier-3, Tier-4 or Tier-5)

Where?

DO subscribe to addon environments (Tier-2+) on a monthly basis

- If you have an enterprise agreement (EA), this can be done through the MPSA
- If you have a CSP engagement, the addon environments can be subscribed for on a monthly basis

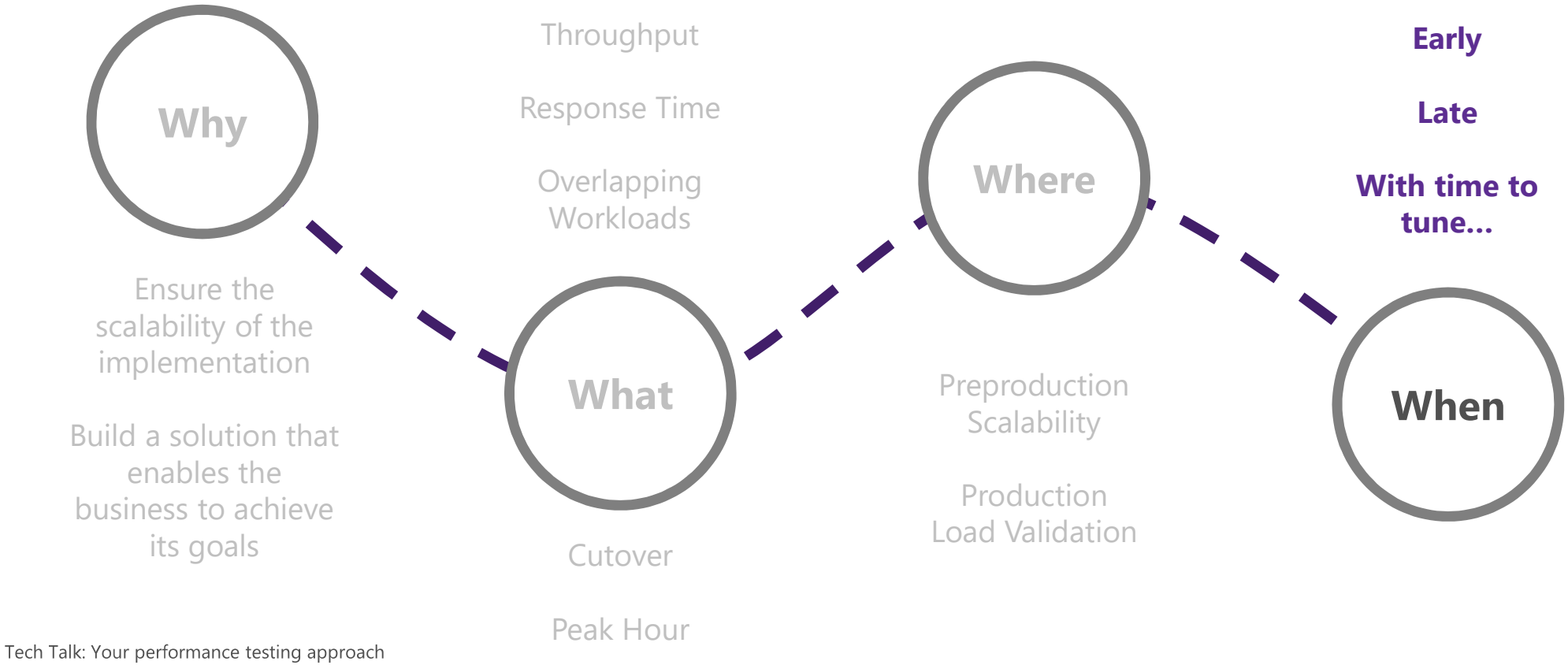
DON'T use the Tier-1 environment

- The Tier-1 environments have a different topology than PRODUCTION (e.g. SQL Server)
- The test results on a Tier-1 would be irrelevant

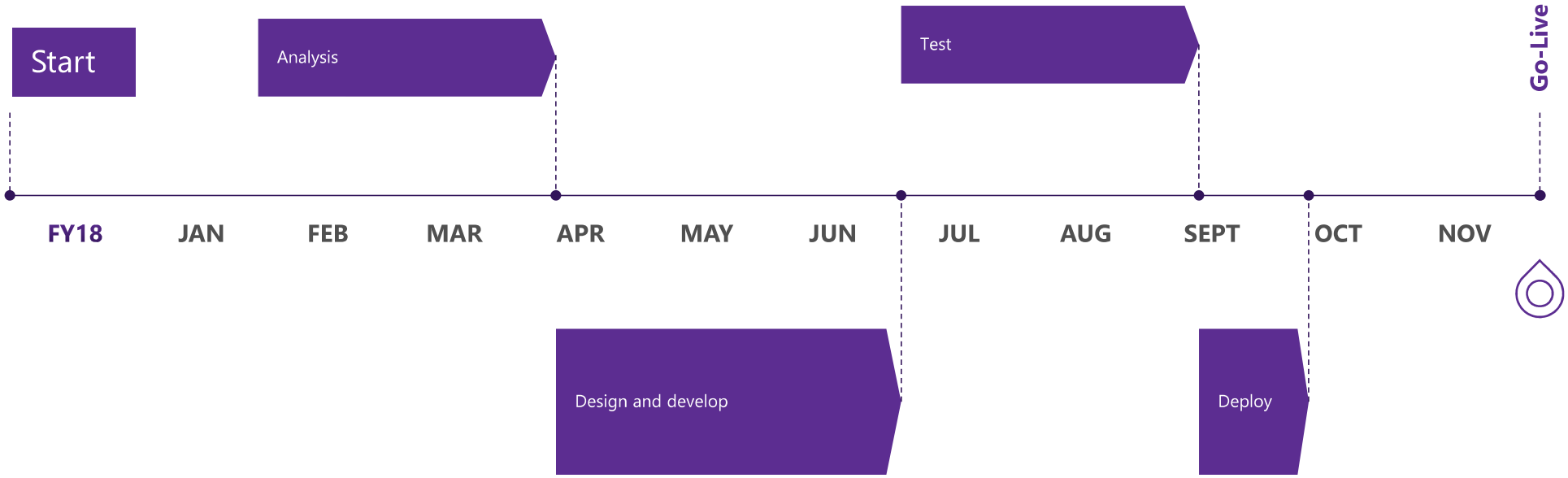
Getting comfortable with the tools

Browser debugger **Fiddler** **Performance timer**	• For the initial investigation (performance timer) • For networking issues (browser debugger / Fiddler)
LCS environment monitoring	• To visualize the activities history on the environment • To analyze the key performance counters of the environment • To view logs and telemetry data (e.g. slow queries)
Trace Parser	• To investigate deeper the performance issues • To analyze the activities at the code and database level
PerfSDK **Visual Studio Load Test**	• To simulate concurrent users • To perform an exhaustive performance benchmark

Performance enhancing practices



When?


Start
Analysis
Test
Go-Live
FY18
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEPT
OCT
NOV
Design and develop
Deploy

When?

Performance throughout the project

GOAL: ensure the final solution is performant, scalable and optimized

Start

Analysis — Capture performance requirements

Test — Execute performance testing and tune

Go-Live

Educate the project team to performance considerations and tools

Educate UAT users to performance issues

FY18 JAN FEB MAR APR MAY JUN JUL AUG SEPT OCT NOV

Environment planning

Performance training

Definition of performance scenarios

Design and develop

Apply performance best practices

Deploy

Mock cutover to validate the maximum workload

When?

Doing performance testing in PRODUCTION before Go-Live?

No, performance testing should happen in a non-production environment because:

- PRODUCTION is not a non-production environment
- As PRODUCTION is released approximately a month before Go-Live, it doesn't give sufficient time to correct any performance issue
- If performance issues are detected in PRODUCTION, they can't be investigated and debugged easily in the environment

What can be done in PRODUCTION is a final round of validation (e.g. maximum workload)

Additional aspects to consider

Responsibilities around performance

Customer

Owns the solution

- Describe clearly the business goals and predict transactional volumes
- Complete the usage profile based on the licenses purchased
- Whenever applicable, optimize the business operations to maximize the efficiency of the solution
- Plan the performance testing and allocate the relevant resources

Partner

Builds the solution

- Understand and document the customer processes
- Capture the performance non-functional requirements
- Build a solution that is optimized/fit-for-purpose that doesn't disrupt the scalability of D365FO
- Provide the expertise to educate the customer on performance testing
- Support the customer for the performance testing execution

Microsoft

Manages the platform

- Leverage both the customer's tests telemetry data and the usage profile to properly size the PRODUCTION environment
- For a performance issue under Microsoft's scope, deliver a solution (e.g. index, hotfix, infrastructure, …)
- For a performance issue outside Microsoft's scope, support the investigation in an advisory case

DO / DON'T

DO

- Apply the latest platform updates and critical hotfixes as they bring significant performance improvements
- Define the workflow pre and post Go-Live for handling performance issues (who triages, who investigates, who opens a ticket if needed, etc.
- Inform the ISVs of your performance testing to benefit from their reactive support in case of issues in their solution
- Provide sufficient information when working with Microsoft Support (see the next slide)

DON'T

- Expect Microsoft to resolve all performance issues (e.g. on incorrect configuration, inefficient business process, inefficient customizations, …)

Being efficient with Microsoft support

DO provide sufficient information to limit emails

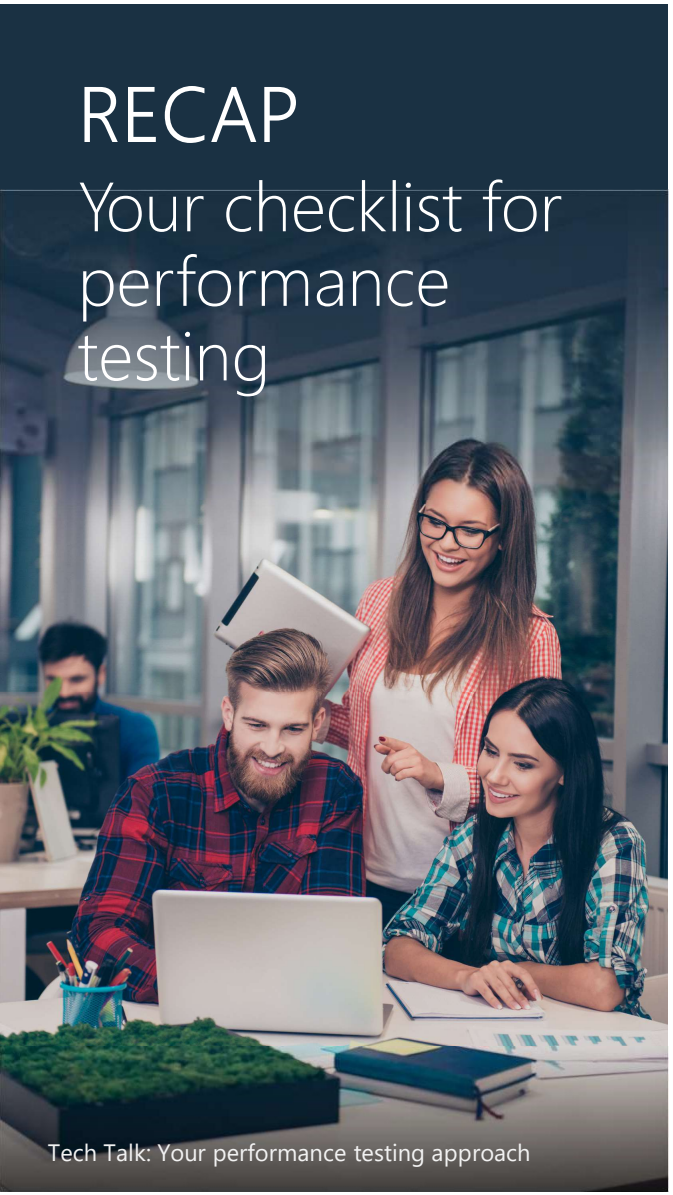
a) What you're trying to achieve
b) Reproduction steps
c) The expected result
d) The actual result
e) The result of your pre-investigation
f) A trace of the performance issue

The more information provided

- The faster the investigation
- The faster the solution (configuration, index, hotfix, other)

RECAP

Your checklist for performance testing



Tech Talk: Your performance testing approach

Use and adjust this baseline checklist

- Did we watch this Tech Talk?
- Have we captured the performance requirements?
- Have we planned performance testing?
- Have we considered the environment for performance testing?
- Do we have sufficient skills to investigate performance issues?
- Have we defined flows for treating performance issues?
- Do we have other parties to involve (e.g. ISV)?
- Do we have a template for Microsoft Support incidents?
- Have we established that our solution is still scalable?
- Are we confident that all our performance critical scenarios will run as expected post Go-Live?

Meet our 2 project teams

SLOW IT

This project team:

- Considers the solution should be performant and scalable by nature
- Considers Microsoft will ensure the implemented solution remains scalable
- Doesn't see the value/necessity of performance testing

FAST IT

This project team:

- Has watched this Tech Talk
- Understands Microsoft makes the standard solution scalable and that additional building blocks (e. g. configuration / customizations) could disrupt this scalability

During the **Analysis** phase

SLOW IT

This project team:

- Doesn't capture the non-functional performance requirements
- Doesn't identify the critical scenarios that are candidates for performance testing
- Doesn't determine which environment to use for performance testing
- Plans UAT, but not performance testing

FAST IT

This project team:

- Educates the project team to focus on performance (non-functional requirements gathering, performance issues during testing, …)
- Defines the environment to use for performance testing
- Allocate time and plan the performance testing

During the **Design** phase

SLOW IT

This project team:

- Focuses on the functional design without considering performance aspects and best practices (e.g. business operations calendar)
- Adopts inefficient patterns at all level (e.g. mono-threaded approach for a heavy batch job, Power BI reports using OData, integration using only OData, …)

FAST IT

This project team:

- Designs an architecture that focuses on business performance
- Adopts performance patterns/practices in the underlying designs (e.g. set-based operations)
- Sends developers to the Advanced Performance Workshop
- Defines workflow for investigating performance issues

During the **Development** phase

SLOW IT

This project team:

- Develops performance costly customizations
- Doesn't include the latest platform/binary updates, waiting for after Go-Live

FAST IT

This project team:

- Defines the flow and guidelines to open a support incident efficiently (especially around performance issue)
- Plans training for the project team on the tools based on their role (e.g. environment monitoring via LCS for the administrator)
- Performs unit tests with maximum load whenever possible
- Includes regularly the latest platform/binary updates and critical binary hotfixes

During the **Testing** phase

SLOW IT

This project team:

- Focuses on UAT and doesn't identify performance issues (e.g. repeated slow form)
- Confirms the solution will work with minimal load but not if it's still scalable

FAST IT

This project team:

- Identifies performance issues 10 performance issues, including 4 critical
  - User interaction hanging then crashing on a ISV form
  - Integration broken, messages queuing
  - Power BI report not loading
  - Heavy batch job taking days instead of hours
- Works with the relevant party (partner, ISV, Microsoft) to resolve the issue
- Receives, applies and validates the correction (configuration, hotfix, index, …)

During the **Deployment** phase

SLOW IT

This project team:

- Performs the mock cutover
- Validates the project team is ready for Go-Live

FAST IT

This project team:

- Performs the mock cutover and validate the maximum workload and overall speed in PRODUCTION
- Reports last minute issues to the relevant parties
- Determines the Support process post Go-Live
- Validates the project team is ready for Go-Live

At Go-Live

SLOW IT

This project team:

- Discovers 10 performance issues including 4 critical
  - User interaction hanging then crashing on a ISV form
  - Integration broken, messages queuing
  - Power BI report not loading
  - Heavy batch job taking days instead of hours
- Requests in emergency a Tier-5 environment for investigating performance issues
  - Suffers the lead time for procuring the subscription

FAST IT

This project team:

- Has limited risk of facing performance issues
- If any performance issue, the tools/processes/actors are in place to investigate and fix efficiently

After Go-Live

SLOW IT

This project team:

- Is communicated that their implemented solution is perceived as a very poor application
- Is on an escalation with the ISV
- Is on an escalation with the external application integrating with D365FO
- Is on an escalation with Microsoft with the Power BI connectors
- Is on an escalation with the partner to rework the batch job process
- Has deprioritized the other performance issues, generating frustration with the end-users

FAST IT

This project team:

- Has limited risk of facing performance issues
- If any performance issue, the tools/processes/actors are in place to investigate and fix efficiently

What is your approach to performance testing now?

Additional resources

Look at the following resources to go further:

- Previous Tech Talk on the performance testing (methodology)
- Advanced performance workshop for developers (example on Dynamics Learning Portal)
- Performance Timer
- Trace parser (take and diagnose a trace)
- Automated testing
- How to create a single user test
- Performance SDK and Multi-user Testing with VSO
- Visual Studio Cloud based testing
- Quick video guide to use the environment monitoring in LCS
- Diagnostics on LCS

Dankie Faleminderit **Shukran** Chnorakaloutioun Hvala Blagodaria

Děkuji **Tak** Dank u Tänan Kiitos **Merci** Danke Ευχαριστώ A dank

Mahalo תודה. **Dhanyavād** Köszönöm Takk Terima kasih **Grazie** Grazzi

Thank you!

>> Next: Q&A

감사합니다 Paldies Choukrane Ačiū **Благодарам** ありがとうございました

谢谢 Баярлалаа **Dziękuję** Obrigado Mulţumesc **Спасибо** Ngiyabonga

Ďakujem Tack Nandri Kop khun **Teşekkür ederim** Дякую Хвала Diolch