

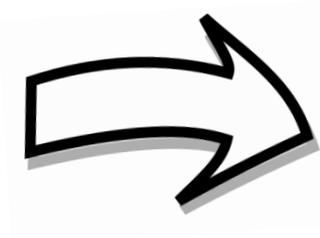
Empower your Teams

How SAP utilizes the Team Foundation Server

Niels Hebling, Gerald Morrison, Dirk Lüttke

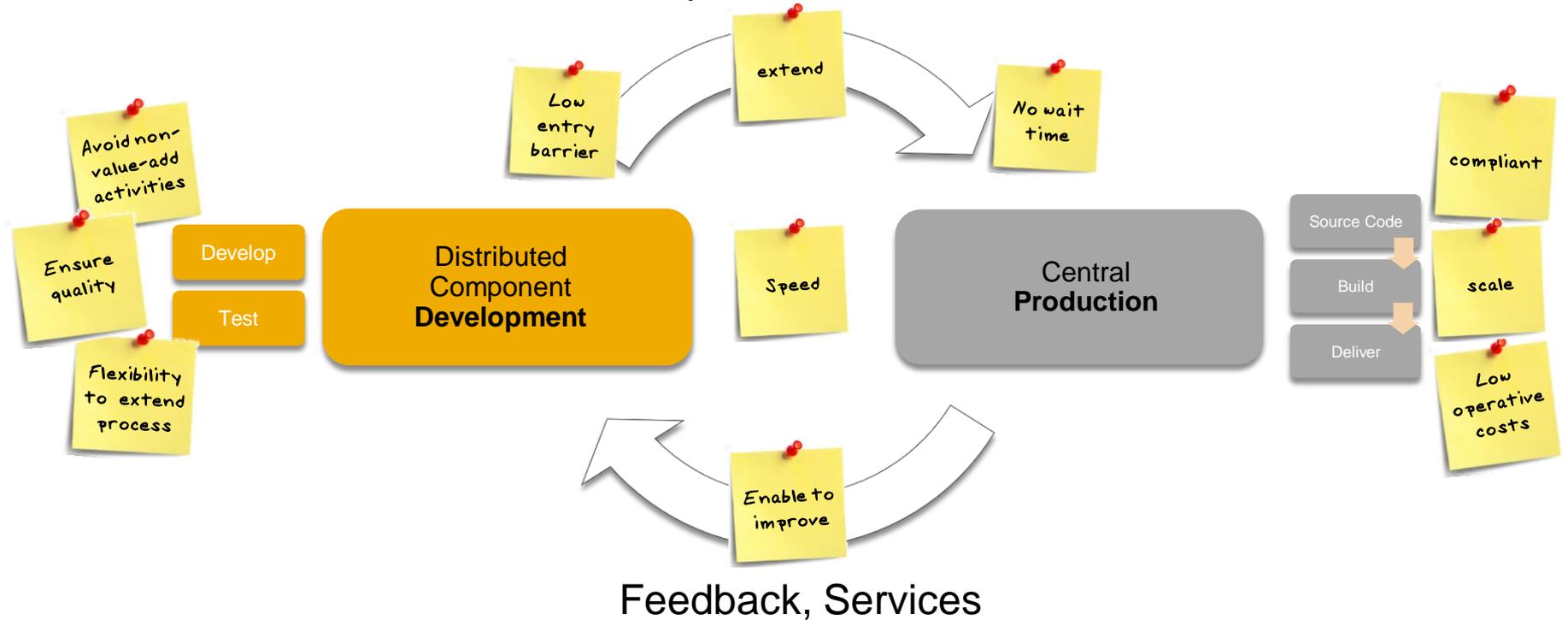


What is it all about?

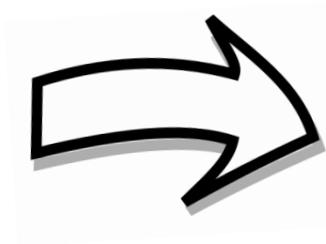


Development Lifecycle in a perfect world

Development Increments



Developer's perspective



Facts

- **SAP Business ByDesign**
 - On-Demand Product for SME & LE
 - ABAP Backend
 - Silverlight Client
 - ~100 .NET developers worldwide
 - ~1.3 Million LoC
 - Several internal tools based on .NET/C#
 - Release cycles 3 month
 - 3 releases to maintain in parallel



In the old days...

- **Visual Studio 2008 / 2010**
- **Perforce Source Control**
- **SAP Japro**
 - SAP's proprietary build environment for Java
 - Extended to support .NET / C#
- **ABAP Backend**
 - Delivery of client runtime via backend



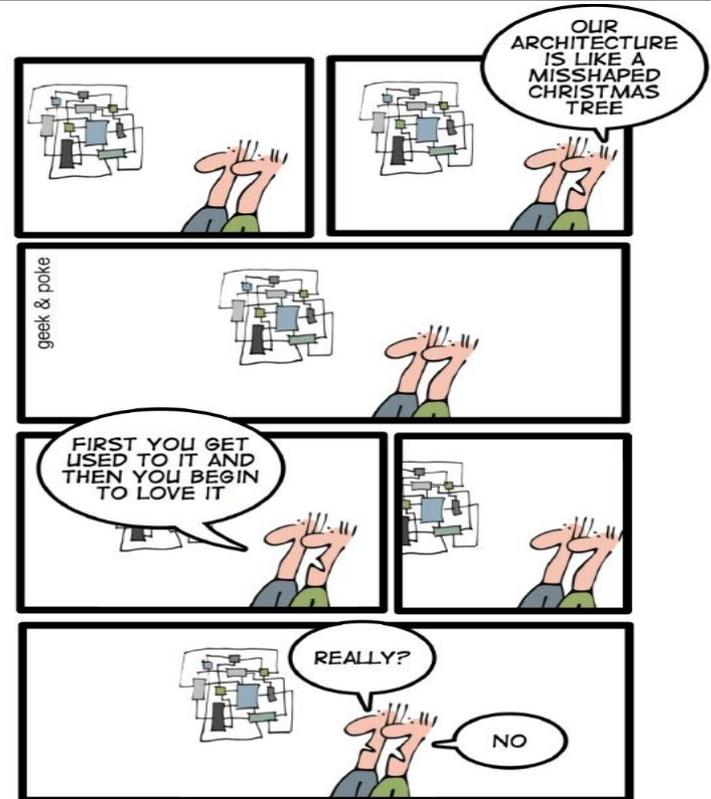
**heterogeneous
environment**

„KISS“ ...or maybe not...

- Dependency Management .NET vs. Java
 - Click-Once deployment
- „Onboarding“ new colleagues not easy
- Feedback loop broke
- Build error problem analysis too complex
- “Googlebarkeit”

Avoid non-value-add activities

Low entry barrier



(Continuous) Integration

- **Testinfrastruktur not integrated**
 - Unit Tests
 - Static Code Checks
- **„Integration“ of source control und build server**
 - Local vs. central build
 - Gated check-in

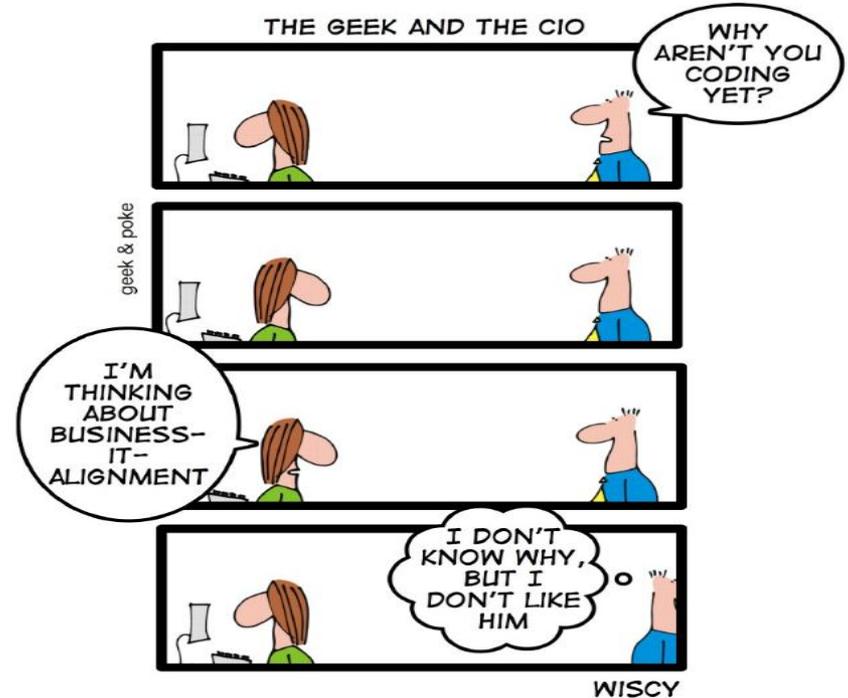


It's not just writing code

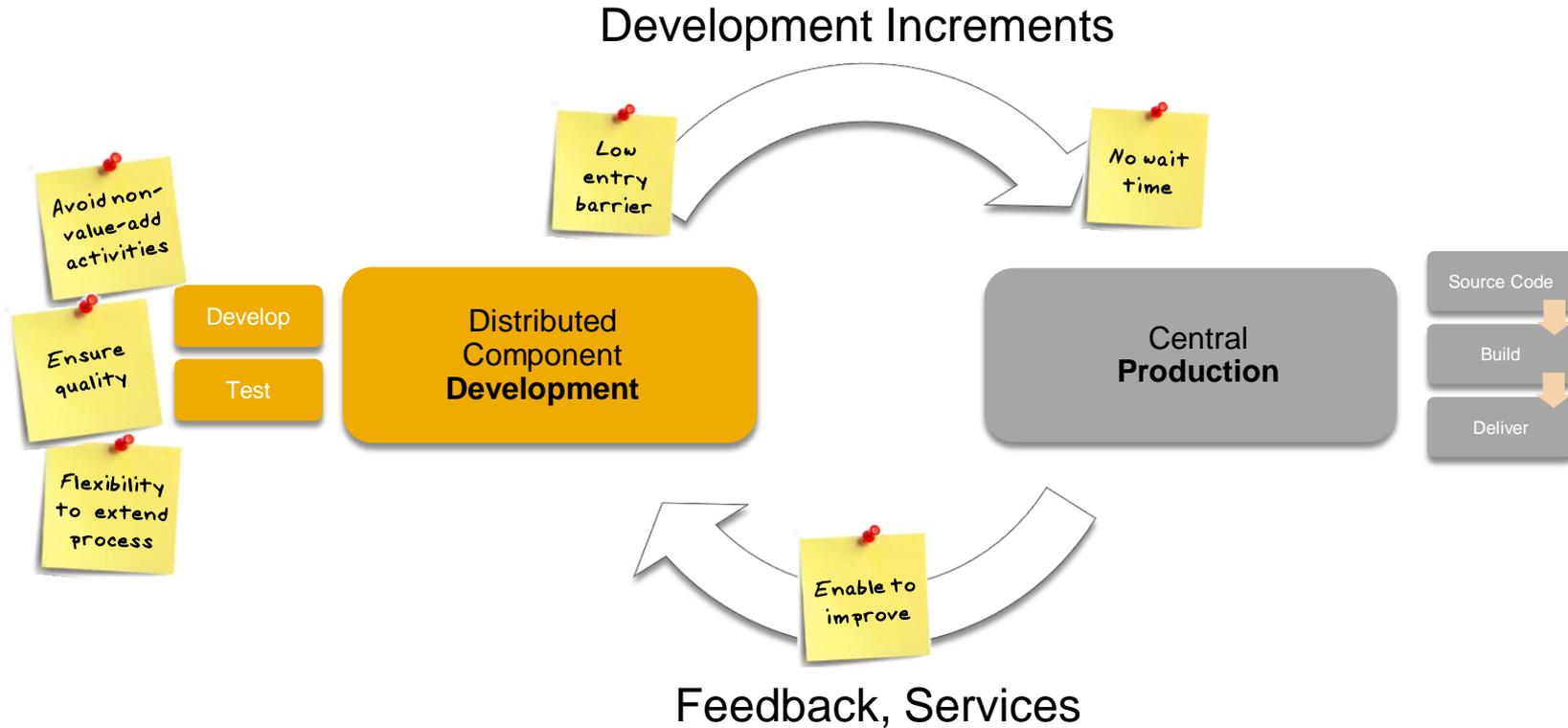
- **Difficult to follow the processes**
 - Tool integration
 - Process/Tool interfaces
- **Expert knowledge not in the teams**
- **Administrative effort increases with each new codeline**

Enable to improve

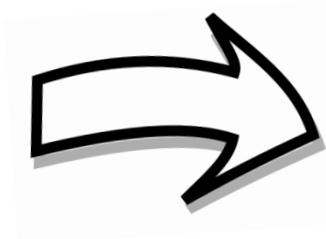
Flexibility to extend process



What makes the difference?



Get connected



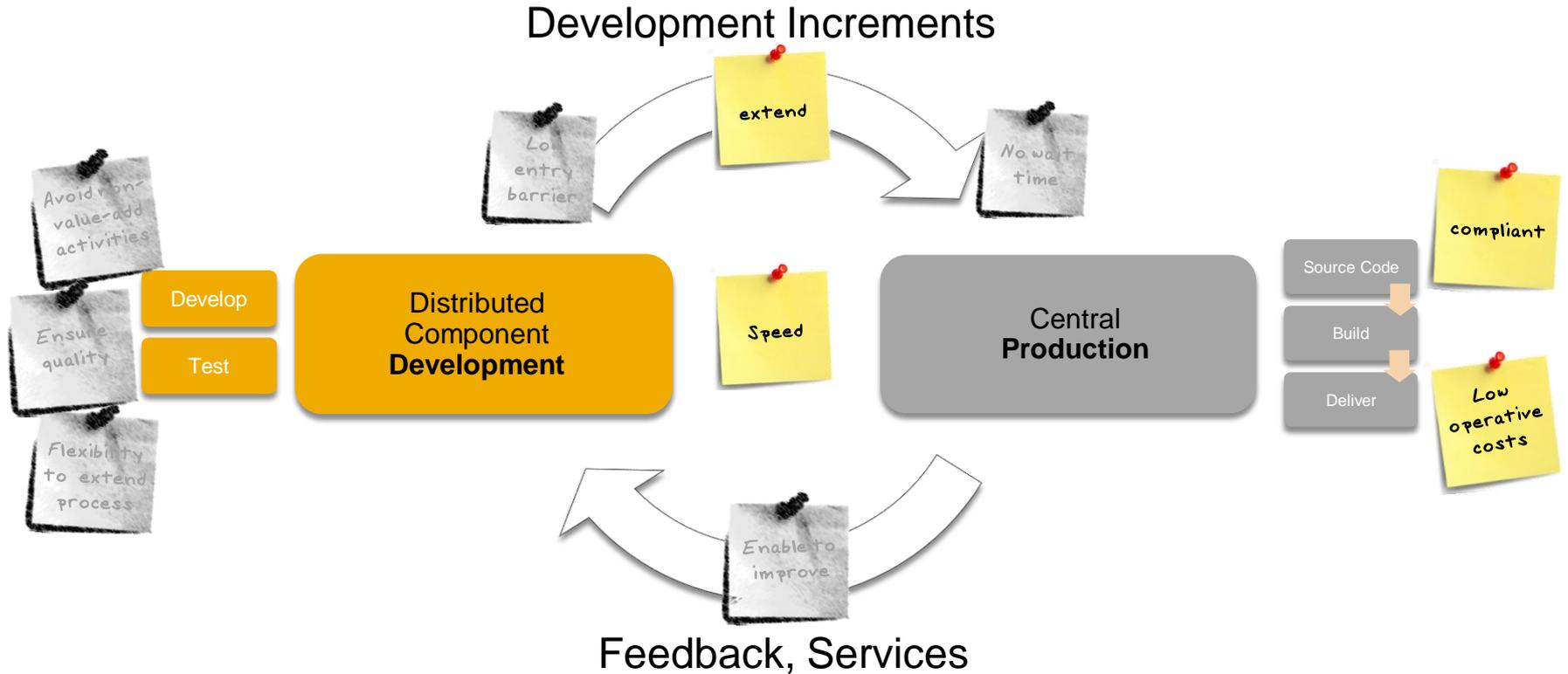
we are
here

How may I help you?

Department 'Cloud RMDA Productization':

- **Offers the Framework for Development to work in:**
 - Release Management
 - Define and set up Environment and Processes for SAP Cloud Products
 - Interface between Development and Service Groups
 - Early Integration of Operation Teams into Development Lifecycle

Topics in my focus



How to...



Speed

... **gain rapid improvement for the development organization?**



compliant

... **be always compliant and able to deliver according to SAP's product standards?**



Low
operative
costs



extend

... **convince central production to invest into another ALM environment?**

On the short and long run



Speed

- **No budget and minimal setup time**
- **No involvement of other departments needed**
- **Addresses top-ranked issues out-of-the-box**
- **Keep developers productive**



compliant

- **Keep current production process alive**



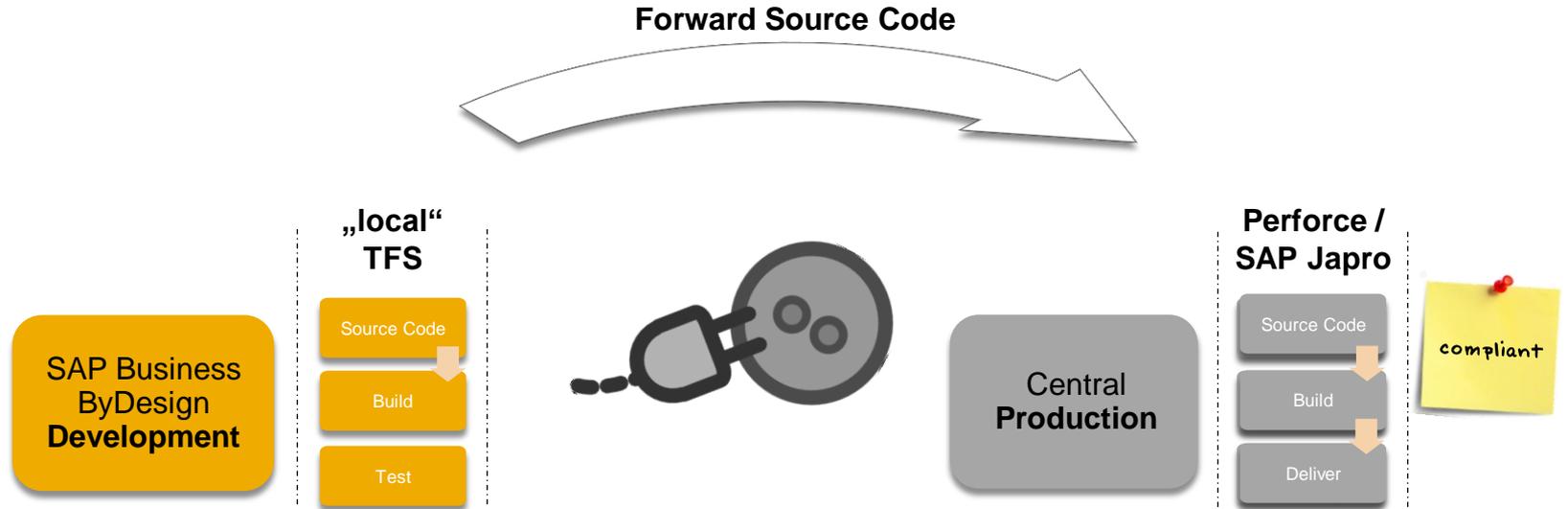
Low
operative
costs

- **Integration into SAP's infrastructure and processes**
- **Functional enhancement**
- **Co-innovation project**



extend

Quick Start: TFS in front of SAP's standard SCM



TFS @ SAP Roadmap

Quick Start

Setup

- 2 persons in ,free-time‘
- Only virtualized hardware

Keep proprietary build

- Translation/Globalization
- Upload build artifacts to ABAP backend

Use TFS in front of central production

- TFS as framework for unit tests and static code checks
- Gated-Checkin in TFS
- Early Detection of integration problems

Development Phase

Infrastructure Management

- Infrastructure Ramp-up of Test landscape - **done**
- Landscape Definition (~2500 Users landscape) - **done**
- Productive Landscape Ramp-up, SLA with IT- **done**

Full Self-Service enablement for Developers

- Self Services for Project Creation, User Management, ... - **done**

Standard Compliance

- Definition of Correction Process - **done**
- Development for Process Coupling (Translation, Correction WB, Signing, ABAP upload, ... - **done**
- Build Results exchange via Common Repository - **in progress**

Operative Ramp-up

- Provide Developer Documentation - **in progress**
- Operation Handbooks - **in process**
- Operation workshops - **done**
- Handover of Process Operations to Sofia Team - **in process**
- Gap Closing Security, leftovers - **in progress**

2010

Oct

2011

Dez

Jan
2012

Mar

Jun

Sep

Nov

2012

Piloting

By Design as first productive user on TFS

- ~100 developers, Exceptions from standard compliance - **done**
- First non-cloud shipment of Windows 8 Apps piloted by Final Assembly - **done**
- Review of Correction Process by Final Assembly - **in progress**

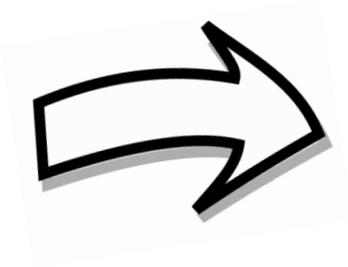
Security Audit

no shipment critical aspects seen by IT Security - **Successfully passed**

General Availability TFS

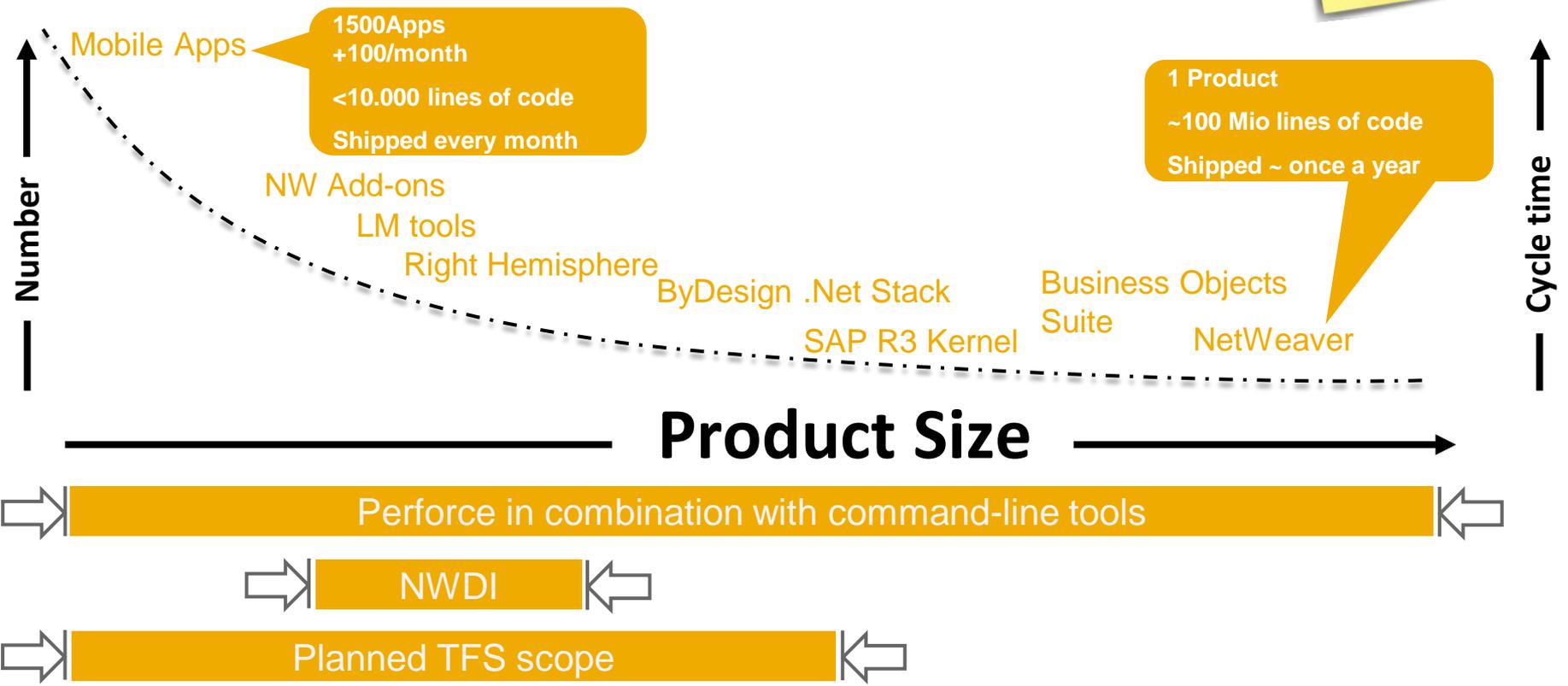
No new .NET projects on Japro - **reached**

Central Production's perspective

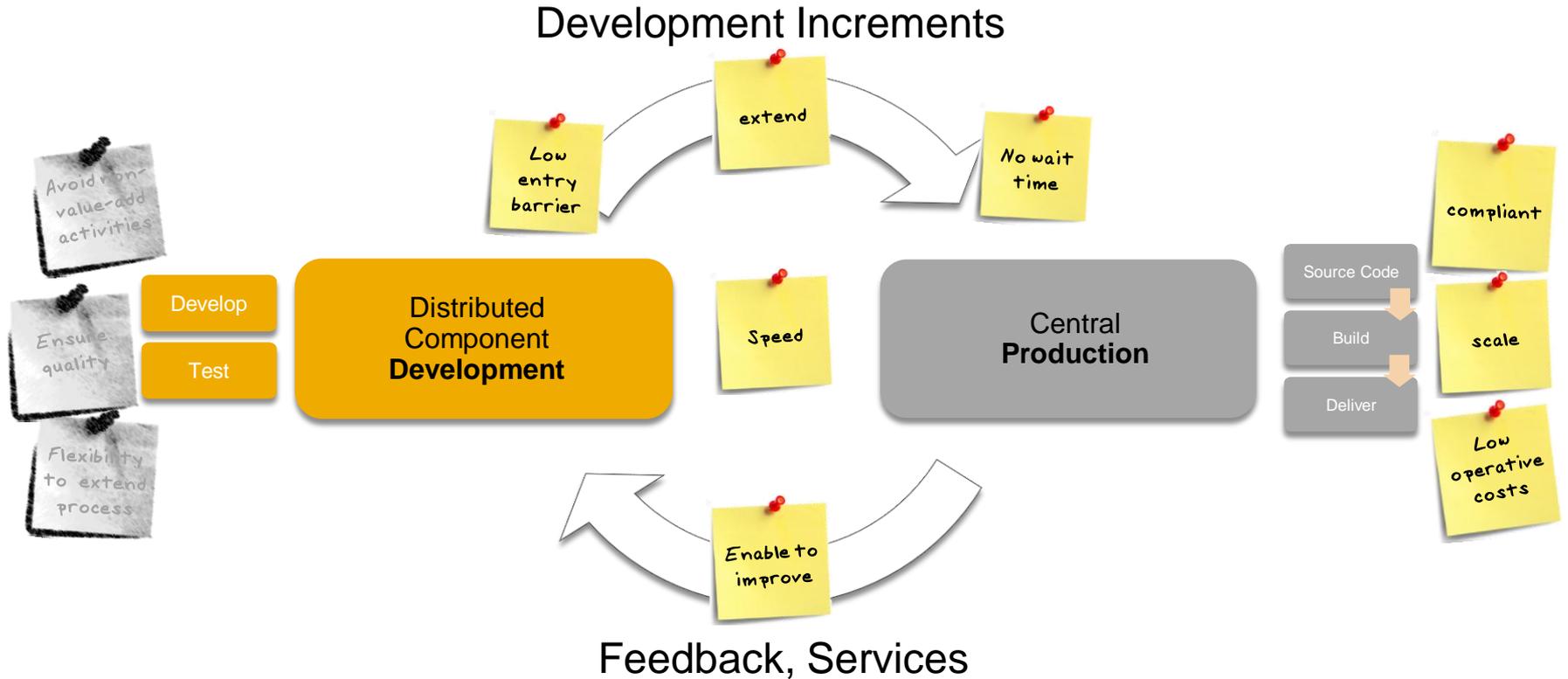


we are
here

Why we love the command line



What is important for me



Why is it important from the central perspective



- **Developers hate the feeling of being dependent**
 - Ticket-free systems, Self Services only
 - Enable decentral enhancements of central systems
- **Keep central costs low**
- **Fast Feedback = Fast Fixing = keep SAP's costs low**
- **SAP in-house processes connected**
- **Reproducible builds**
- **Security built-in**

TFS – our common Point of view



Speed

- Fast TFS Ramp-up
- End-to-end integration, single point of entry



No wait time

- Full Self Service enablement
- Don't feel to be at expert's mercy
- Self descriptive problems or „Google is your friend“



extend

- Possibility for decentral enhancements
- No "silo thinking" and local improvement only

Extensibility

- Still missing extension points in VS 2012 for several TFS entities (e.g. work items)
- UI of build result summary cannot be extended

TFS – our common Point of view



Avoid non-value-add activities

- Many out-of-the-box functions
- Main focus on development, secondary services easy to consume
- Check-in policies not centrally distributable

Ensure quality

- Gated Check-in
- Static Code Checks
- Integration of Unit Testing
- No central reporting/history of static code check results

Low entry barrier

- Very fast acceptance by developer community
- Full integration into developer workplace
- Big User Community (MSDN, Google)

TFS – our common Point of view



compliant

- Relatively simple to connect SAP-specific processes

Security

- No version management for build definitions
- Missing authorization roles on Build Controller level

Dependency Management

- Handover between components with different lifecycles in not sufficiently supported
- Back traceability from artifact to source not clear

Low operative costs

- Administration support is nice!

- Local dependencies to global assembly cache -> central installation of libs required
- Very tedious tasks that always requires central turnaround.