

# IBM Measured Capability Improvement Framework (MCIF) Überblick

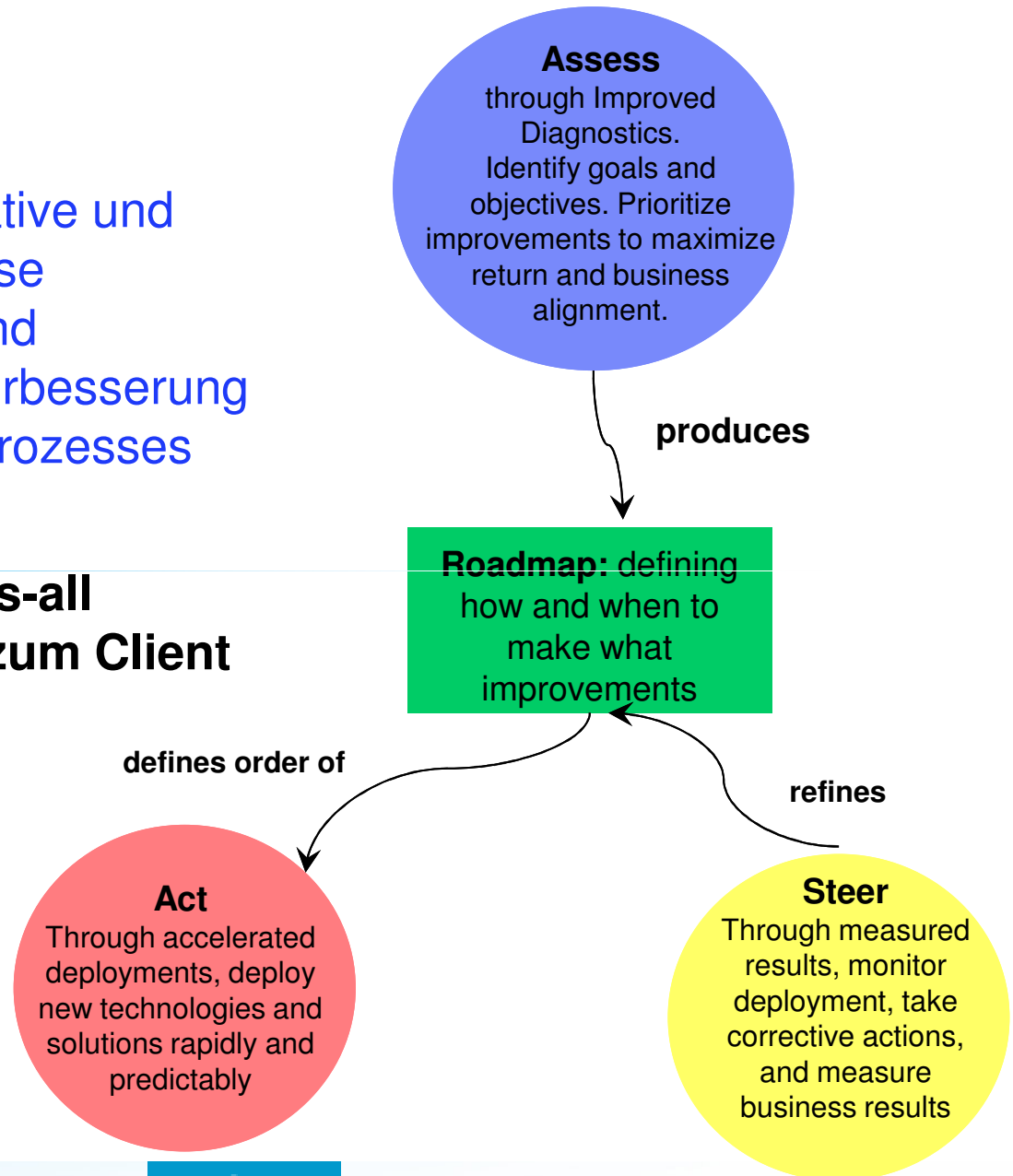
**Rational.** software

*Dejan Mihajlovic*  
*Senior Managing Consultant*  
*IBM SWG Austria*

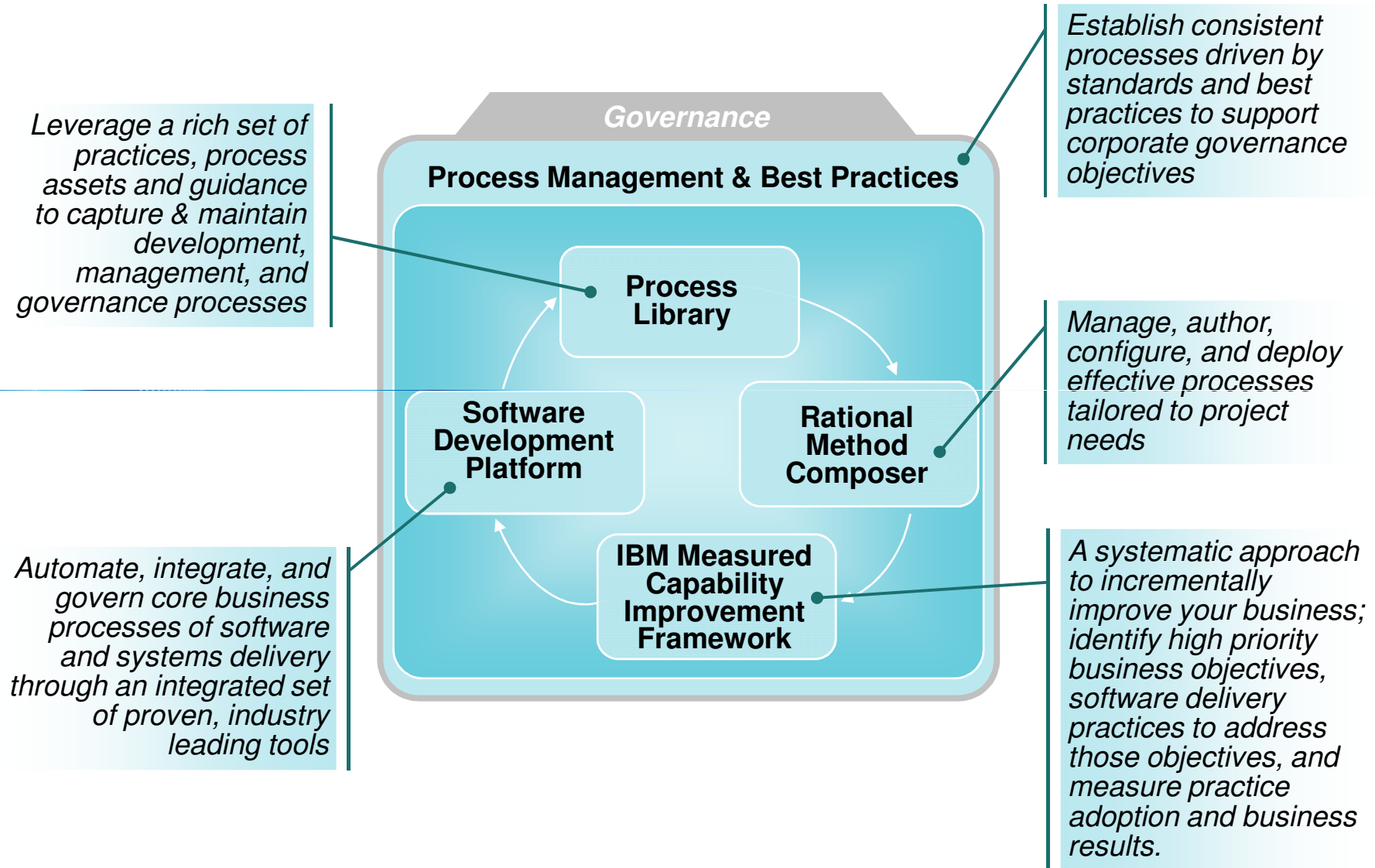
[dejan\\_mihajlovic@at.ibm.com](mailto:dejan_mihajlovic@at.ibm.com)  
**+43 6646185879**

# Was ist MCIF

- ist eine systematische, iterative und formalisierte Vorgehensweise (unterstützte durch Tools und Services), um messbare Verbesserung des Softwareentwicklungsprozesses zu erreichen
- **MCIF ist nicht one-size-fits-all**  
recipe: es ist angepasst zum Client Context



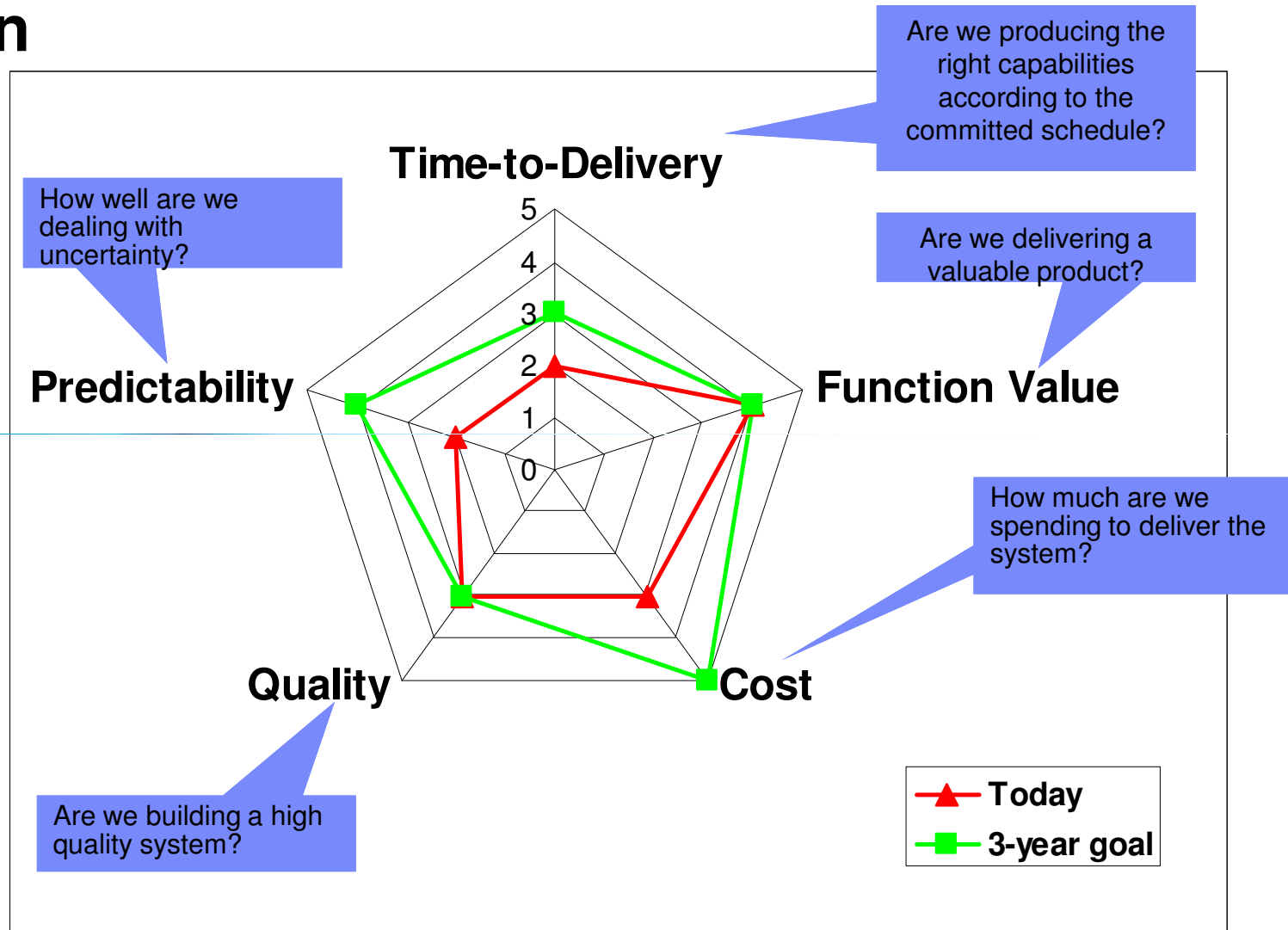
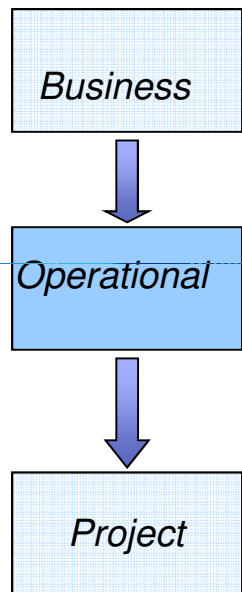
# IBM Prozessmanagement - Überblick



## Folgende 5 Bereiche können bei der MCIF Assessment berücksichtigt werden

Dimensionen	Heute	3-Jahres Ziel
<b>Time-to-Value</b>	Fix pack wird jede 2 Monate publiziert	Die Frequenz der Fix Pack Releases soll erhöht werden (einmal in Monat)
<b>Business Value</b>	Nur 20% der Bank Transaktionen sind On-Line	Erhöhen den Anzahl der On-Line Banking Transaktionen zu 80%
<b>Cost</b>	Wartungskosten sind 75% der gesamten IT Kosten	Reduzierung der Wartungskosten auf 65%
<b>Quality</b>	20% Defects bleiben bei der SW Lieferung	Reduzierung der Defects auf 10%
<b>Predictability</b>	70% der Projekten sind unterschätzt	Reduzieren der Anzahl der unterschätzen Projekten auf 30%

# Verfolgung der Verbesserungen für die 5 Bereichen



## MCIF kann in folgenden Schritten durchgeführt werden

- IBM® Quick Assessment
  - Ziel: “Etablieren eine RoadMap so schnell wie es möglich ist“
- IBM® Health Assessment
  - Ziel: “In-depth Analysis des Ist-Zustands, Kontext, und der Verbesserungszielen, um eine RoadMap für die Verbesserung zu definieren”
- IBM® Self-Check
  - Ziel: “Involvieren die Team Mitglieder während des Änderungsprozesses und effektiv das Wissen über die Praktiken übertragen.”

## MCIF Quick Assessment bringt folgende Ergebnisse












- Visualisiert Stärken und Schwäche des Softwareentwicklungsprozesses
- Erlaubt eine initiale Verbesserung-Roadmap mit minimalem Aufwand zu erstellen
- Bietet leichten und schnellen Zugang zu IBM Erfahrungen in Verbesserung des Softwareentwicklungsprozesses
- Dokumentiert die Prioritäten der Verbesserungen und Symptome in den Problembereichen
- Zeigt, in welcher Reihenfolge sollten die Verbesserungen vorgenommen werden, um die Aufwände zu optimieren

## Folgende Empfehlungen können aufgrund von Quick Assessment gemacht werden

Question Number	Relevance	Solution (Tactic)	Symptoms
1		Improve lifecycle collaboration	Our team is well integrated across functions (requirements, code, test, and geographies). Collaboration is great and information flows seamless

Practices	Strategies
<p><b>Iterative Development:</b>  <a href="http://rpository.svl.ibm.com:3000/wikis/all_practices/index.htm#practice.mgmt.iterative_dev.base/guidances/practices/iterative_development_CD1297B8.html">http://rpository.svl.ibm.com:3000/wikis/all_practices/index.htm#practice.mgmt.iterative_dev.base/guidances/practices/iterative_development_CD1297B8.html</a></p> <p><b>Release Planning:</b></p>	<ul style="list-style-type: none"> <li>~ Improve Software Delivery Efficiency (Cost)</li> <li>~ Improve Team Collaboration (Time, Value, Predictability)</li> <li>~ Increase Defect Prevention (Quality)</li> </ul>

## Vordefinierte Metriken erlauben die Verbesserungen objektiv zu messen

- [-]  Measurements
  - [-]  Core Measurements
    -  Build Health
    -  Defect Density at Customer Ship
    -  Iteration Burndown
    -  Iteration Velocity
    -  Test Coverage of Requirements
    -  Test Execution Status
  - [-]  Supplementary Measurements
    -  Blocking Defects
    -  Blocking Work Items

**Maßeinheiten, Regeln und Analyse-Möglichkeiten sind als Vorschlag vordefiniert**

## Folgende Input Informationen werden für Quick Assessment gebraucht

Question	Relevance	Symptoms	Answer
			(1 - 5)
5,2	Important	Build results are effectively communicated to team members and managers.	2
5,3	Important	Tests (a.k.a. smoke tests) are automatically triggered once builds are done.	1
6	Important	We always know who changed what and why, making it easy to manage changes across teams, sites, and product components.	3
6,1	Important	It is easy to find out in which build a specific change request was included and It is easy to know which changes were included in any given build.	4
6,2	Important	Builds are baselined which enable us to associate a stable source code configuration with your build.	2
6,3	Important	We have no problems managing and supporting multiple versions (variants, revisions) of our applications at the same time.	5

## Ein typisches MCIF Projekt Szenario

- **Quick Assessment**

- Identifizierung der Herausforderungen und Problemdomains
- Definition der notwendigen Praktiken

- **Pro identifizierten Problemdomain:**

- Deep Dive mit Hilfe von ausgewählten Fragen
- Identifizierung von Measurements für die ausgewählten Praktiken

- **Self-Check Enablement**

- Das Entwicklungsteam soll lernen, wie man sich selbst „überwachen“ kann
- Implementierung der Praktiken, die den Entwicklungsprozess verbessern sollen

A woman in a blue business suit is sitting at a desk with a laptop. She is smiling and raising her right fist in a celebratory gesture. The background is a bright, modern office with large windows. The text "Thank You" is overlaid in the center of the image.

# Thank You