akquinet

Kaizen in der Software-entwicklung

Stefan Roock
stefan.roock@akquinet.de
Henning Wolf
henning.wolf@akquinet.de

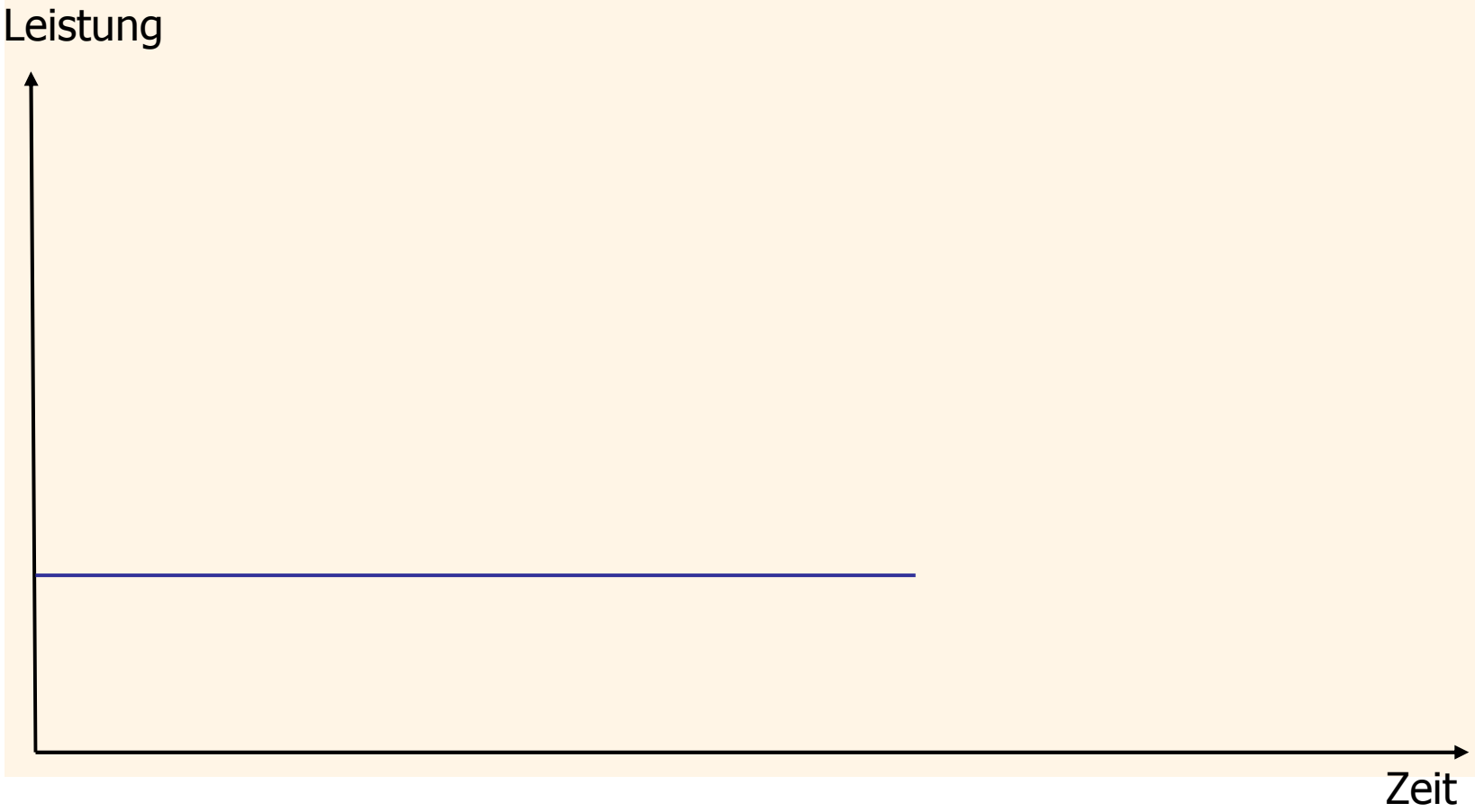
it-agile

Agile Alliance
CORPORATE MEMBER

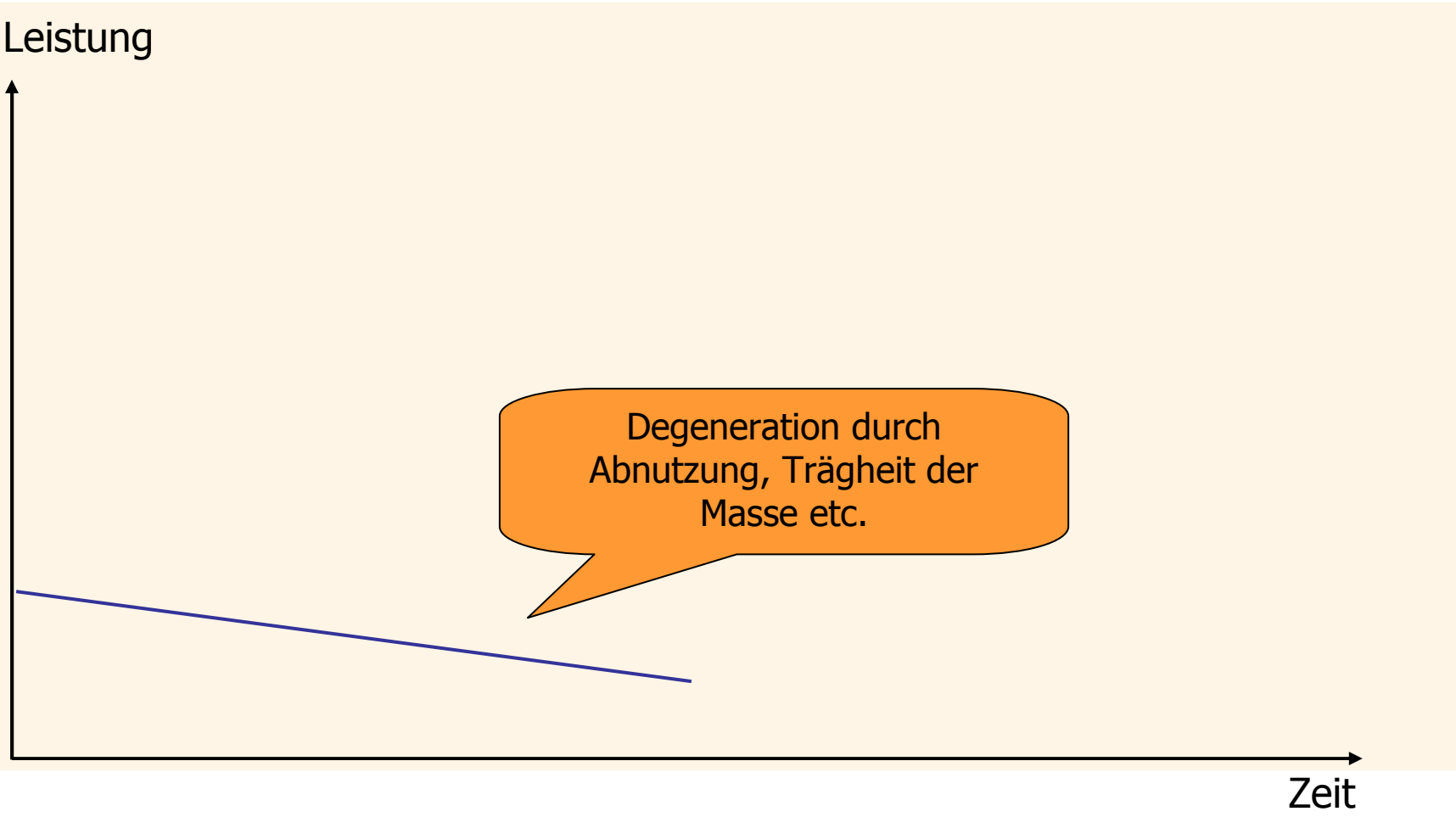
Kaizen

- Nur ein guter Prozess liefert gute Produkte.
- Schwächen im Produkt werden durch Schwächen im Prozess verursacht.
- Prozessverbesserung führt zu Produktverbesserung.
- Kein Tag ohne Verbesserung.

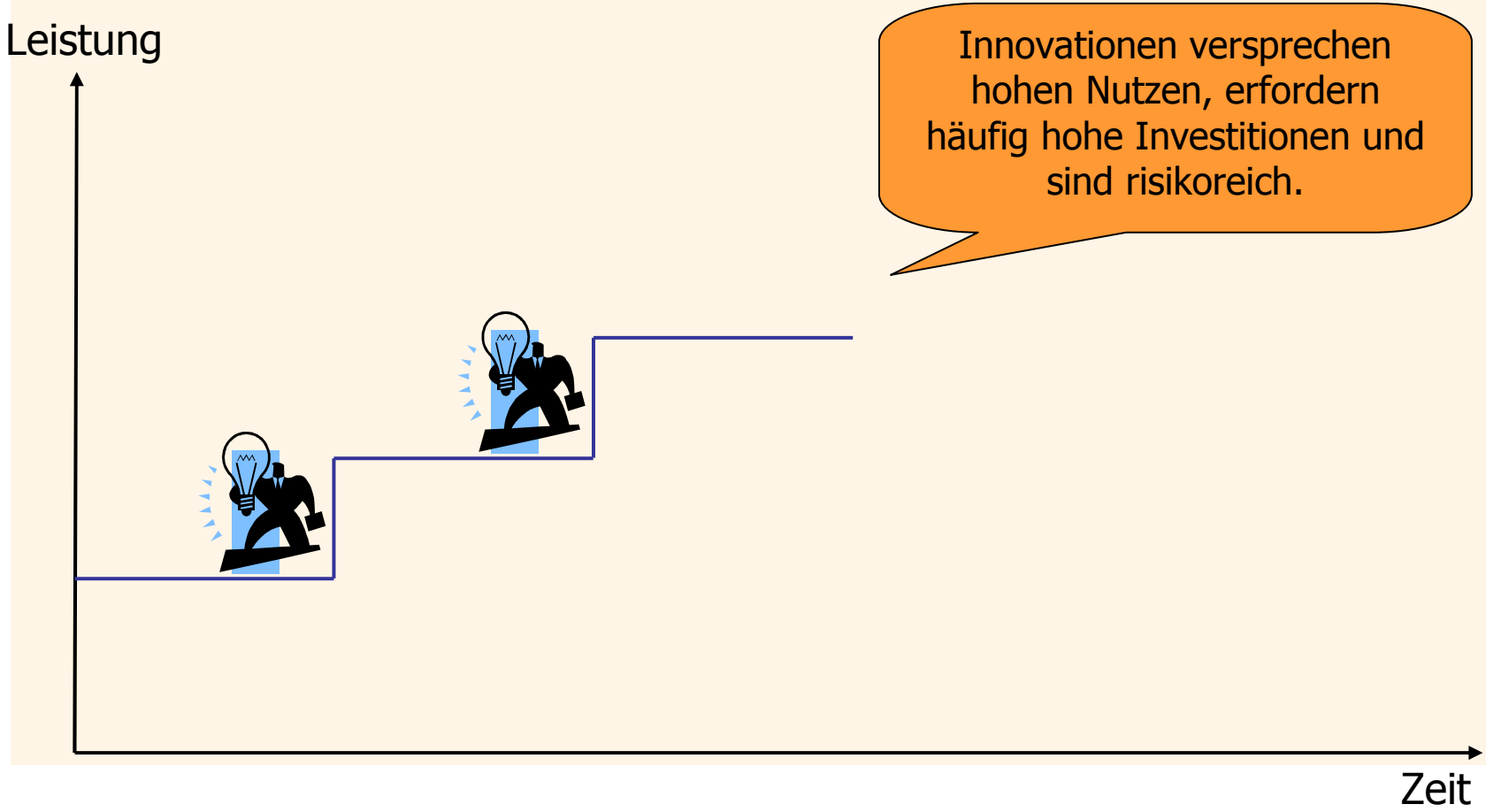
Verwaltung: Theorie



Verwaltung: Praxis

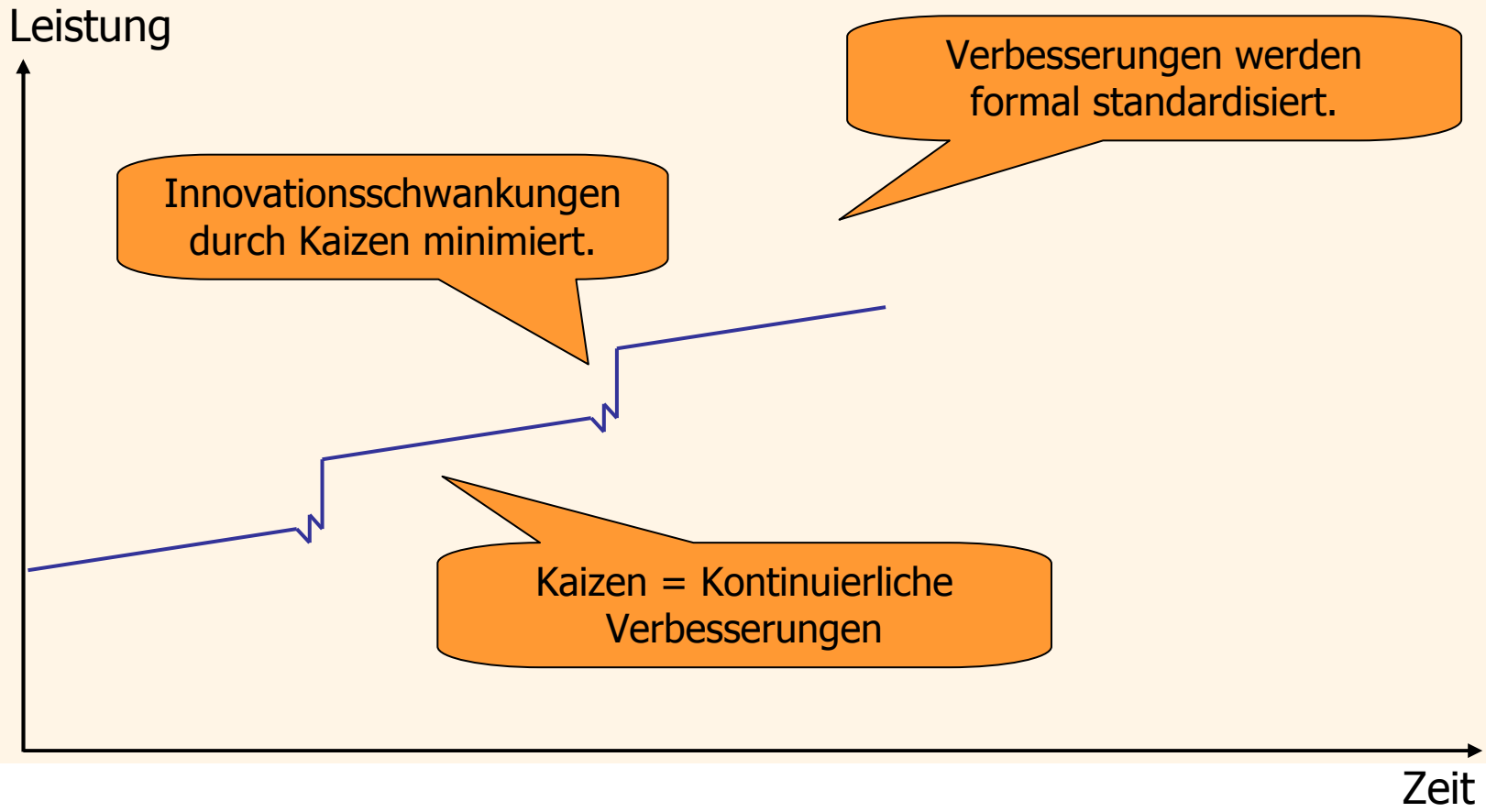


Innovation: Theorie

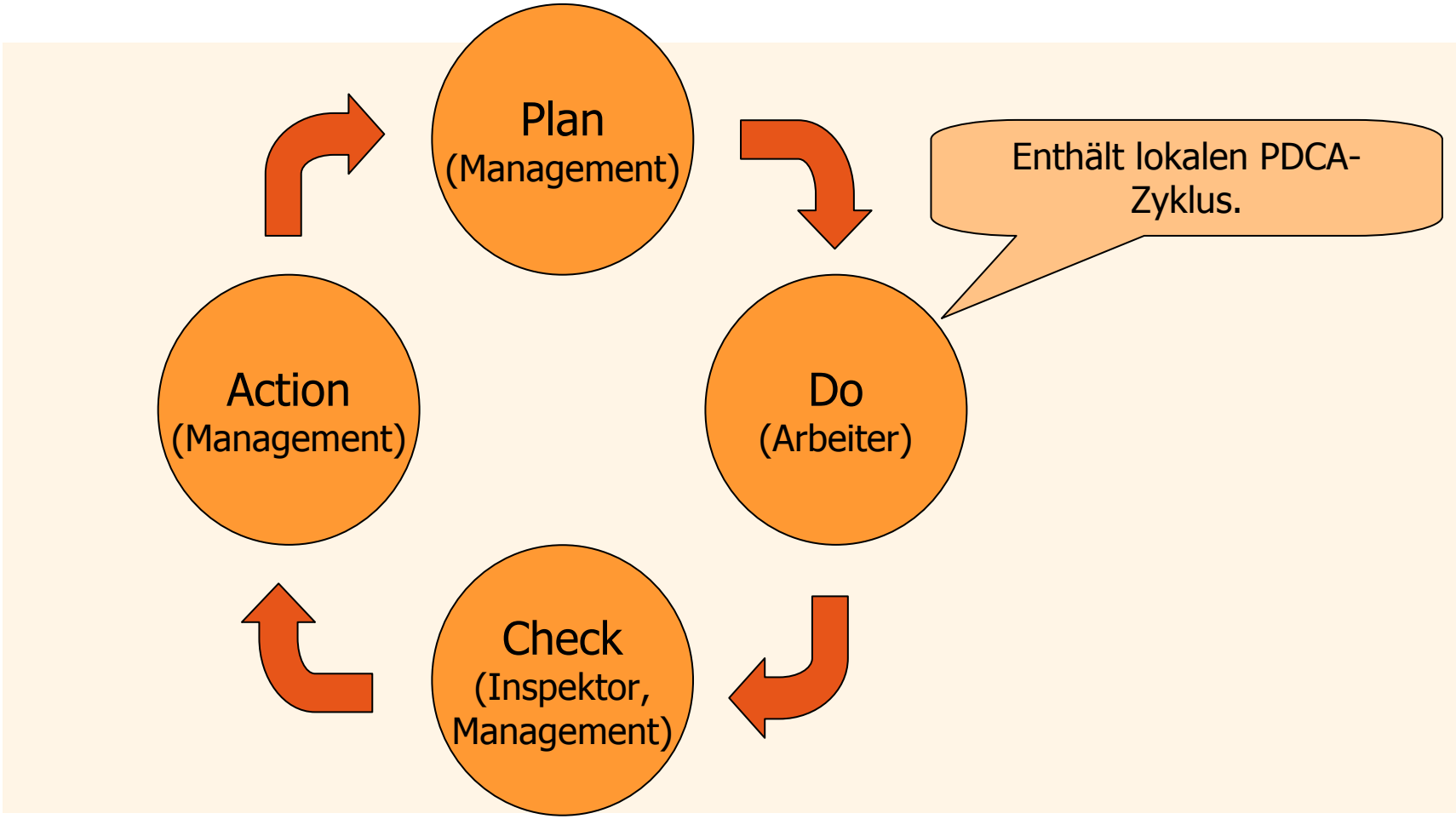


Innovation: Praxis

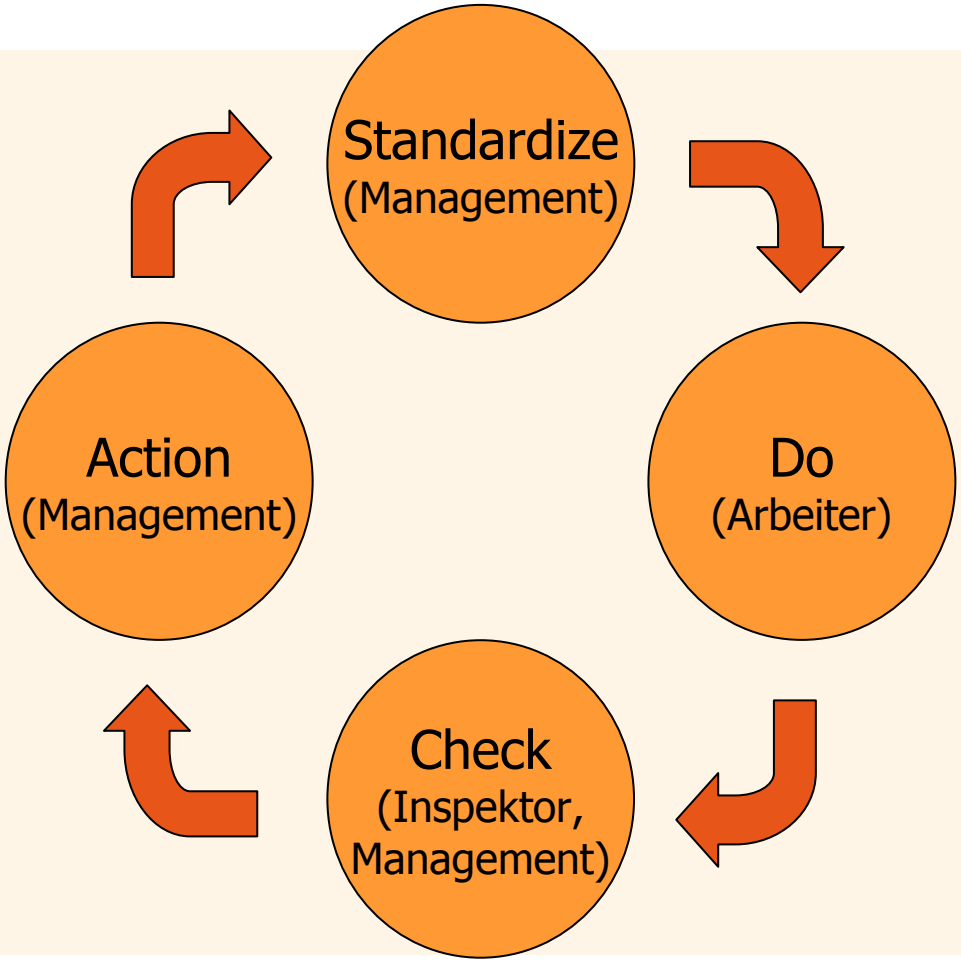
Kaizen


Leistung
Verbesserungen werden formal standardisiert.
Innovationsschwankungen durch Kaizen minimiert.
Kaizen = Kontinuierliche Verbesserungen
Zeit

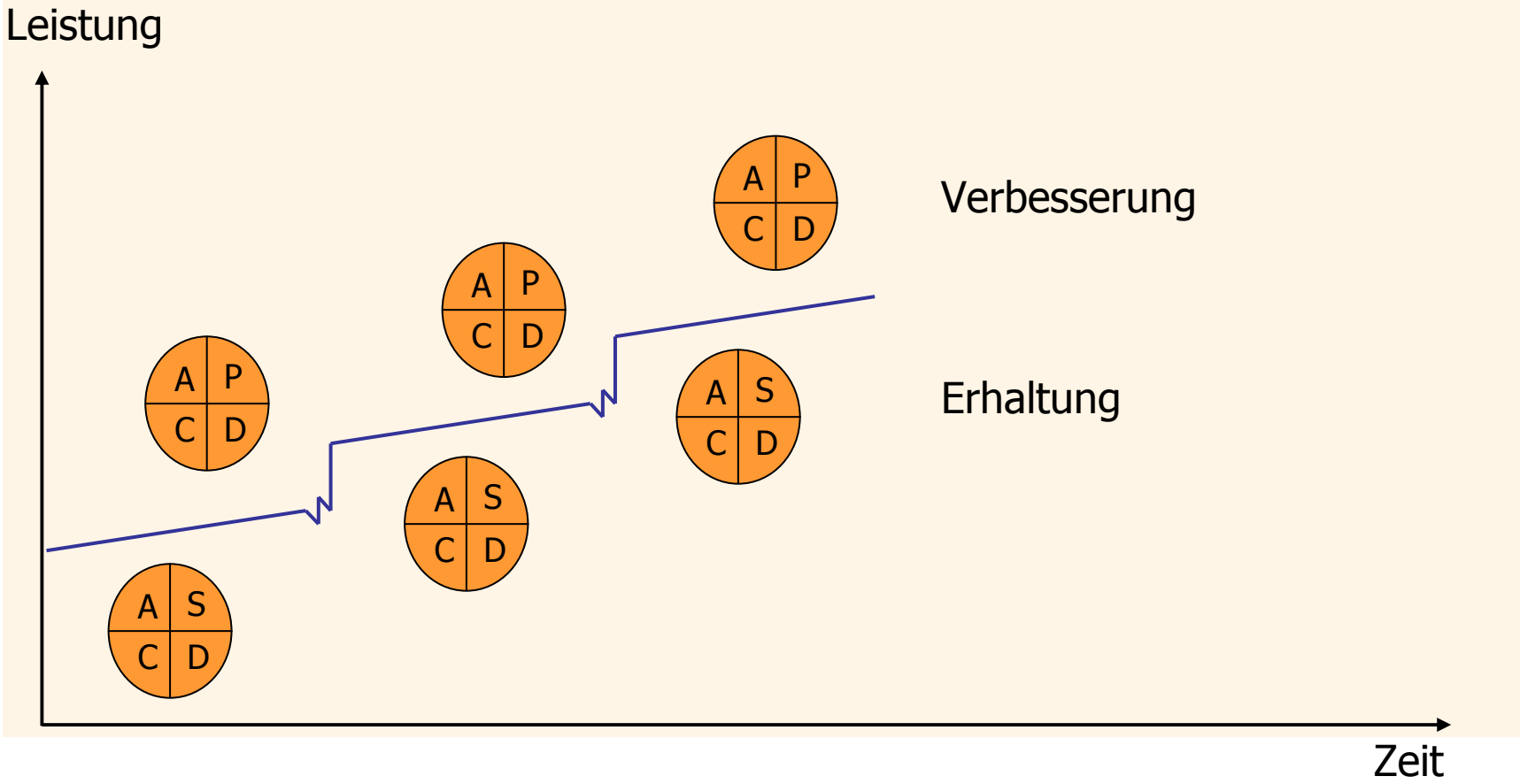
Verbesserung: PDCA-Zyklus



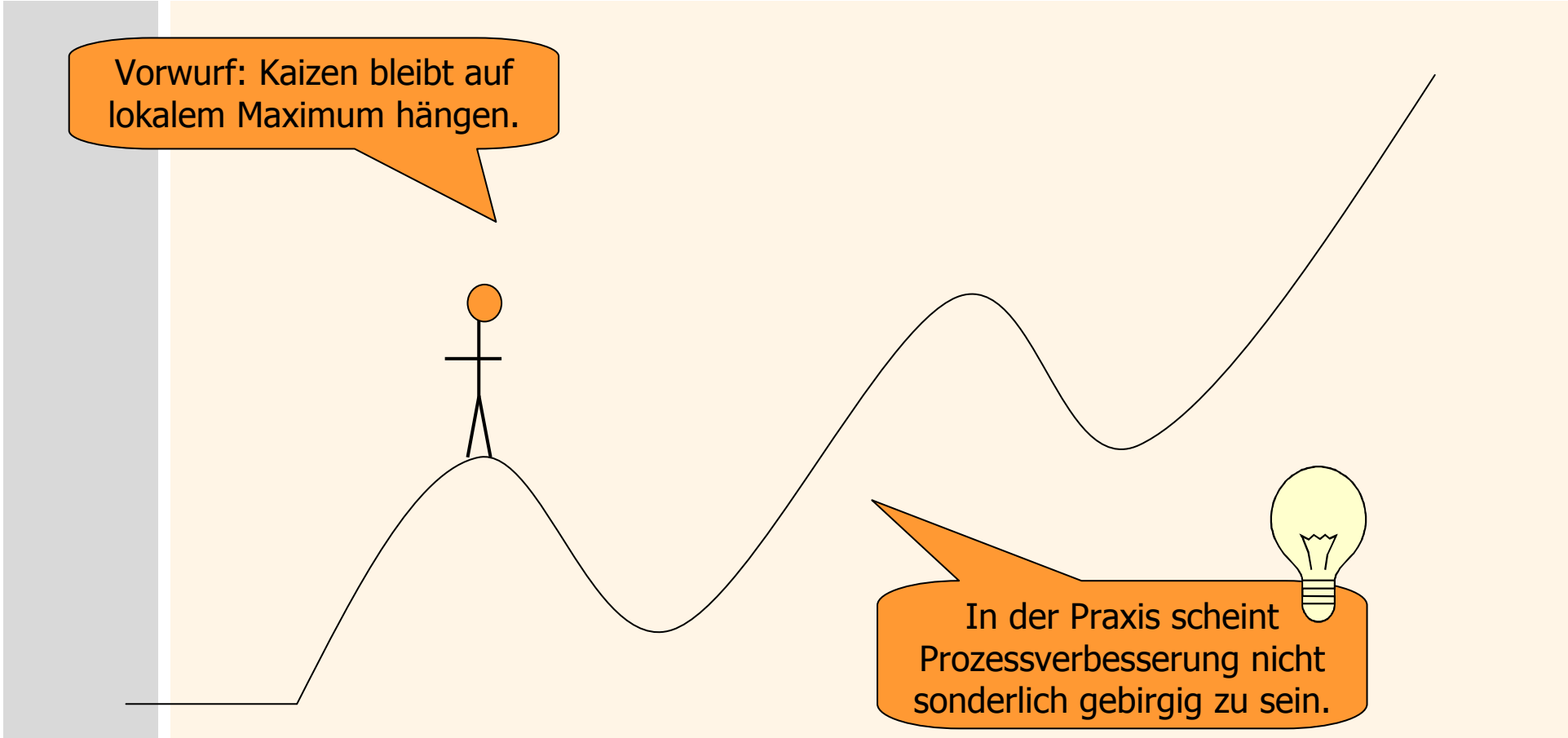
Erhaltung: SDCA-Zyklus



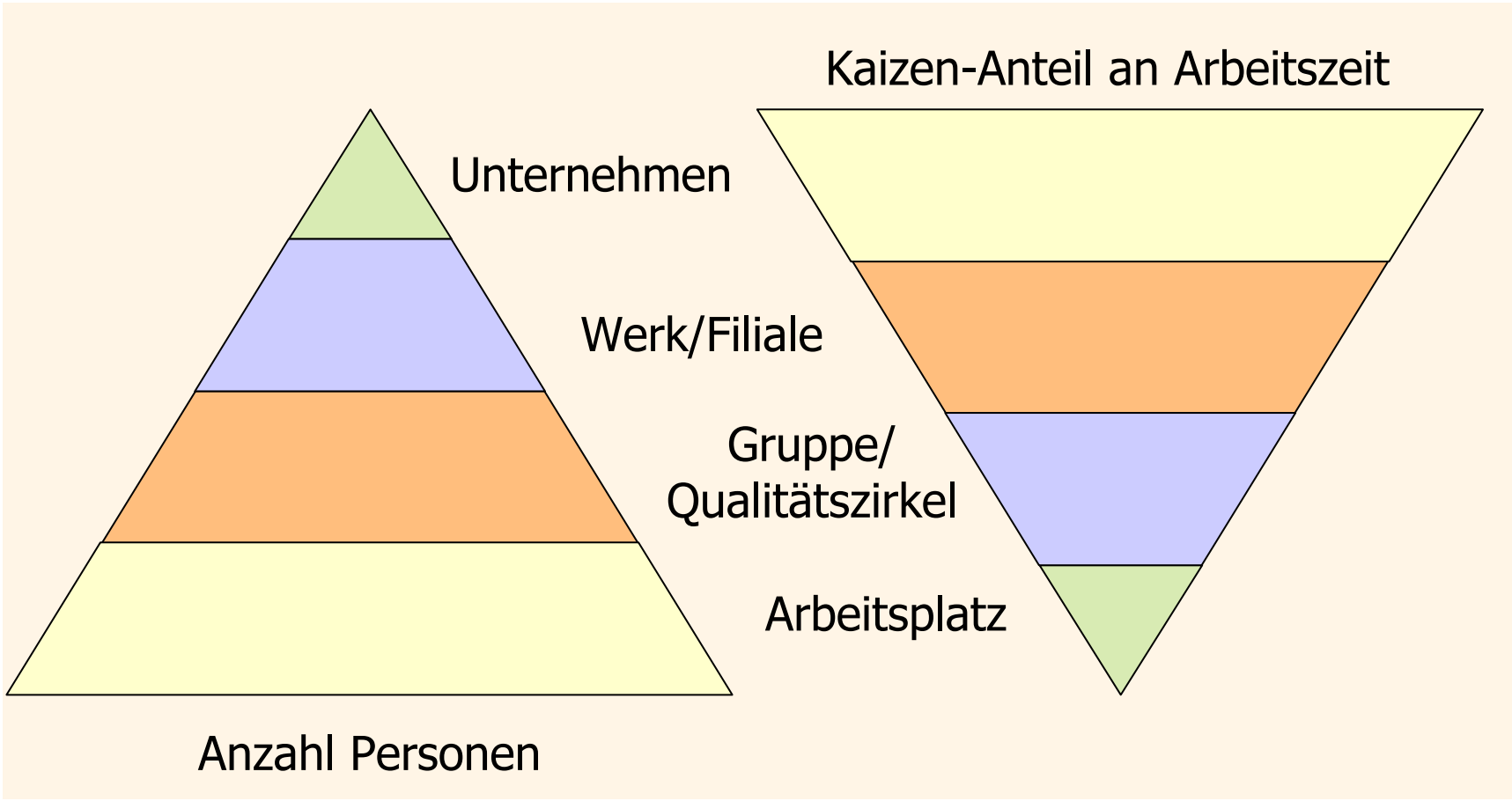
Kaizen


Leistung
A P C D
A P C D
A P C D
A S C D
A S C D
A S C D
Verbesserung
Erhaltung
Zeit

Kaizen und das Gebirge



Ebenen von Kaizen



Grundprinzipien Kaizen 1/2

- Verbesserung = Qualität
- Deutlicher Fokus auf Prozessaspekte.
- Kleine bis sehr kleine Verbesserung mit minimalem Risiko und minimalen Investitionen.
- Keine detaillierte Kosten-Nutzen-Analyse. Kosten der Analyse übersteigen meist die Kosten der Umsetzung.
- Niedrige Zugangshürde. Überall liegen Formulare für Verbesserungsvorschläge aus und existieren sind Briefkästen zum Einwerfen der Formulare.
- Verbesserungen werden sehr schnell umgesetzt (kurze Feedbackzyklen).

Grundprinzipien Kaizen 2/2

- Bewährte Verbesserungen werden zum Standard (Arbeitsanweisungen).
- Verbesserungen werden geschult.
- Manager werden an den Verbesserungen gemessen, die ihre Mitarbeiter vorgeschlagen haben.
- Mitarbeiter bekommen Anerkennung für ihre Verbesserungsvorschläge.
- Verbesserungsvorschläge können jederzeit zu jedem Thema eingereicht werden. Meist gibt es aber Kampagnen, die vom Management ausgehen und die Verbesserungsbemühungen fokussieren.

Qualitätszirkel

- Mitarbeit in Qualitätszirkeln ist freiwillig.
- Unterschiedliche Arbeitszeitregelungen: Arbeitszeit, Überstunden, Freizeit
- Qualitätszirkel bestehen in der Regel aus Arbeitern und Meistern. Manager nehmen in der Regel nicht teil.
- Qualitätszirkel suchen sich ihre Themen selbst aus.

Kaizen in Japan 1/2

- Bekanntestes Ergebnis: Toyota Production System
- Japanische Manager investieren 50% ihrer Arbeitszeit in Kaizen.
- Bei Aisin-Wamer hat im Jahr 1982 jeder Mitarbeiter im Schnitt 127 Verbesserungsvorschläge eingereicht (Gesamtunternehmen: 223.986). Davon wurden 99% umgesetzt (nationaler japanischer Durchschnitt: 76%).
- Canon hat im Jahr 1983 390.000 Verbesserungsvorschläge umgesetzt. Das hat 1,08 Mio. $ gekostet und 94 Mio. $ eingespart.

Kaizen in Japan 2/2

- Bei Nissan Motors werden alle Verbesserungsvorschläge akzeptiert, die mind. eine 1/100 Minute sparen. Hinweis: 0,6 Sekunden sind ein Handgriff, ein Schritt oder eine Körperdrehung.

Kaizen und agile Methoden

Agile Methoden haben Grundaspekte von Kaizen

- Fokus auf Einfachheit
- Kleine Schritte (z.B. beim Testen und Refactoring)
- Anpassung des Prozesses an die Gegebenheiten (z.B. durch Retrospektiven)
- Betonung des Menschen
- Nutzung der Kreativität aller Beteiligter

Aber

- Agile Methoden bedeuten nicht Kaizen.
- Standardisierung spielt in agilen Methoden kaum eine Rolle.

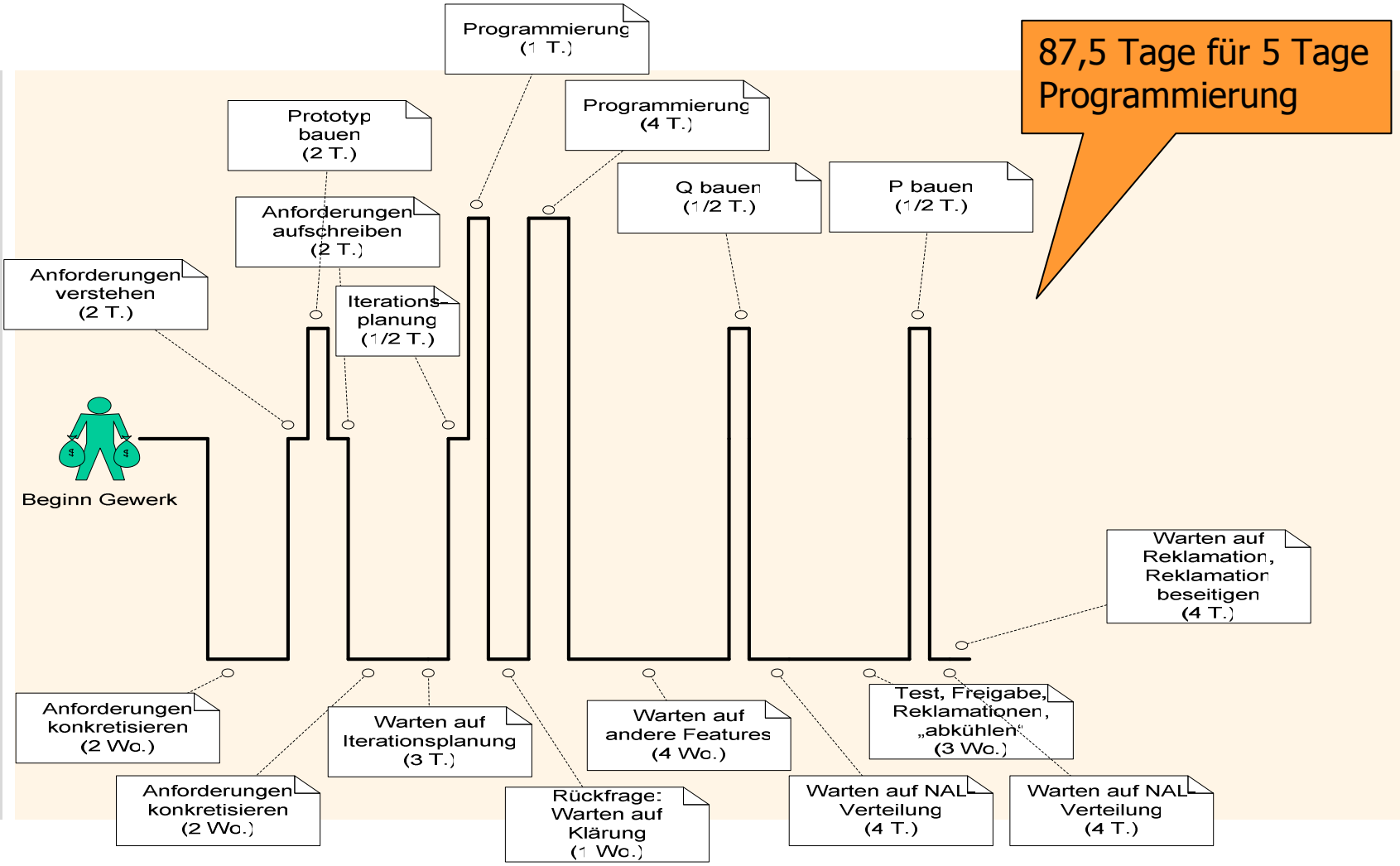
These

- Agile Methoden können durch Kaizen weiterentwickelt werden.

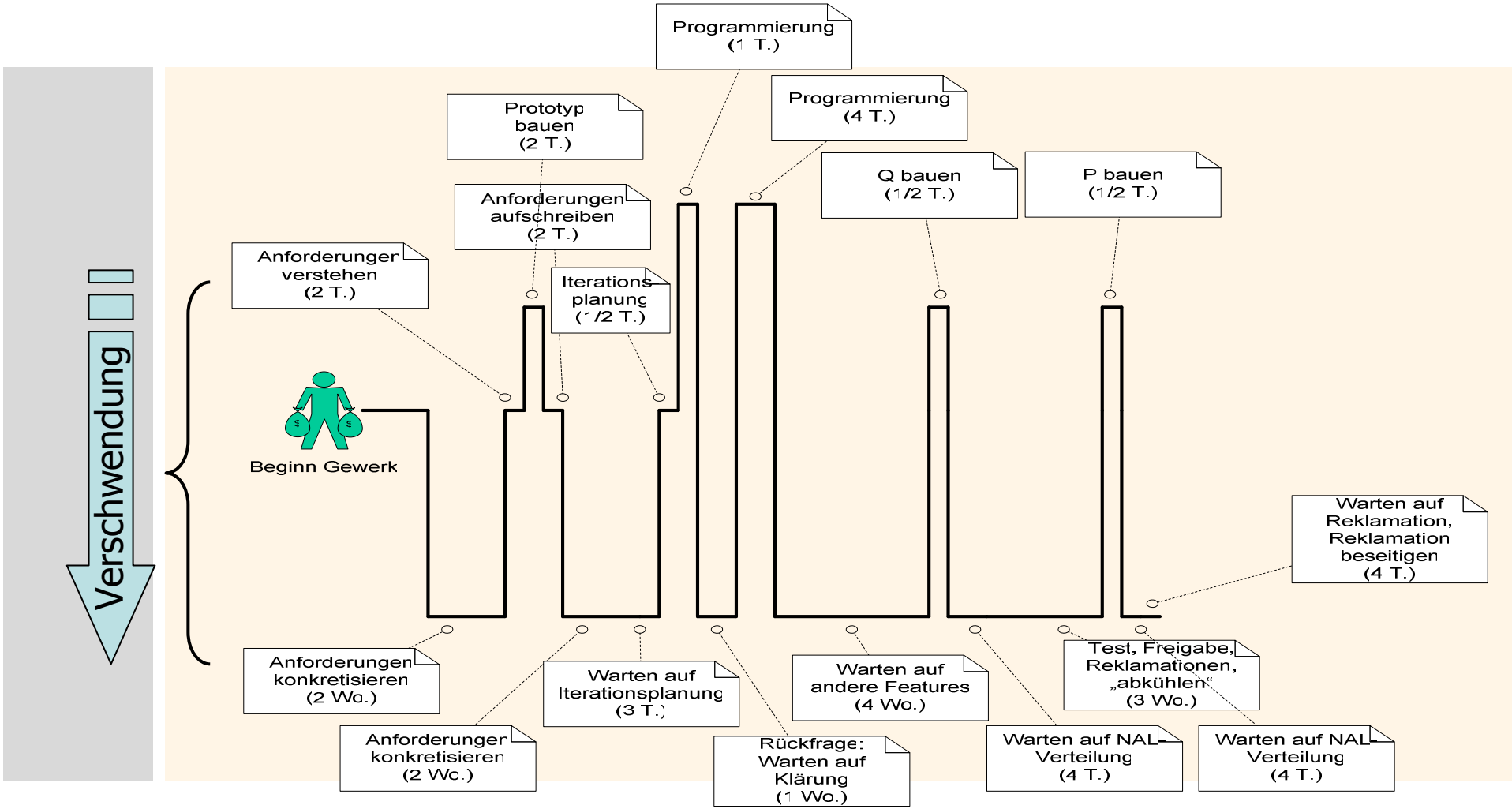
IT-Kaizen

- Deutlicher Fokus auf Prozessaspekte. Kaizen ist keine Diskussionsrunde über Softwarearchitektur.
- Projektleitung / Management definiert Kampagnen für Kaizen.

Ziel Nummer 1: Durchlaufzeiten reduzieren



Feind Nummer 1: Verschwendung



Typische Kampagnen in der IT

- Reklamationen (Bugs) reduzieren.
- Verlässlichere Schätzungen liefern.
- Kontinuierlichere Entwicklungsgeschwindigkeit schaffen.
- Einarbeitungszeit für neue Entwickler reduzieren.
- Änderbarkeit des Systems erhöhen.
- Krankheitsrate reduzieren.
- Pull-Off-Zeiten reduzieren.
- Kontextwechsel / Störungen reduzieren.
- Missverständnisse mit dem Kunden reduzieren.
- Deployment-Zeiten reduzieren.

Verbesserungen am Arbeitsplatz

- Probiert jeder Entwickler zuerst bei sich aus.
- Wenn sich die Verbesserung bei ihm bewährt, erstellt er einen formalen Verbesserungsvorschlag.
- Papier-Formulare für Verbesserungsvorschläge liegen in jedem Büro aus.
- Der Projektleiter entscheidet binnen einer Woche über den Verbesserungsvorschlag und gibt dem Einreicher Feedback:
  - Ablehnung mit Begründung
  - Akzeptanz und Standardisierung für alle Entwickler.
  - Eskalation, wenn der Vorschlag weitreichender ist.

Qualitätszirkel

- Treffen sich wöchentlich für zwei Stunden am Rande der regulären Arbeitszeit (z.B. 17-19 Uhr).
- Arbeiten jeweils für mehrere Sitzungen an einem größeren Thema (immer Prozessverbesserung).
- Erstellen je Thema einen kurzen Abschlussbericht.
- Dienen nicht der Weiterbildung der Teilnehmer.
- Bestehen aus 3 bis 6 Teilnehmern.
- Können mit oder ohne Projektleiter, Kunden etc. stattfinden.
- Qualitätszirkel finden sich selbst und melden sich selbst als solche bei der Projektleitung an.

Projekthandbuch

- Standards werden im Projekthandbuch festgehalten.
- Mitarbeiter werden im Projekthandbuch geschult.
- Änderungen am Projekthandbuch werden deutlich mitgeteilt und ggf. auch geschult.

- Themenbereiche für das Projekthandbuch
  - Kaizen-Vorgehen
  - Code-Conventions
  - Konventionen für Verzeichnisstrukturen
  - zu verwendende Konfiguration der Entwicklungsumgebung
  - Entwicklungsprozess mit Verantwortlichkeiten und Kommunikationswegen

Motivation zur Mitarbeit

- Besonders gute Verbesserungsvorschläge werden ausgehängt.
- Projektleiter / Manager geben Anerkennung für Verbesserungsvorschläge und Mitarbeit in Qualitätszirkeln.
- Es werden Angebote für Schulungen in Kaizen gemacht.
- Öffentliches Tracking über Verbesserungsvorschläge
  - Anzahl von Verbesserungsvorschlägen
  - Anzahl umgesetzter Verbesserungsvorschläge
  - Durchschnittliche und maximale Umsetzungszeit
  - Kosten/Nutzen über alle Verbesserungsvorschläge

Beispiel für Kaizen-Formular

Verbesserungsvorschlag	Nr.
Datum	Autor
O Akzeptiert. Standard seit	
O Eskaliert an	
O Abgelehnt, weil	
Datum	Projektleiter

Erfahrungen mit IT-Kaizen im Projekt 1/2

- Vorstellung Kaizen/IT-Kaizen für gesamtes Team kam sehr gut an
- Schon während der Vorstellung in Übungen ca. 100 Verbesserungsvorschläge und Ideen für 5 Qualitätszirkel
- „Projekthandbuch“ und Standardisierung ungewohnt für agile Entwickler (und erstmal auch leicht bis schwer abschreckend)
- Projekt bestand aus 4 Teilteams, in den Teilteams gibt es durchaus unterschiedliche Herangehensweisen; deshalb sind viele Verbesserungen oder Standardisierungen, die für das gesamte Projekt gelten sollen, nicht auf jedes Team gleich gut anwendbar
- Papierformulare kamen nur bedingt gut an, letztlich wurde doch ein Bugtracking-System eingesetzt für die Vorschlagsverwaltung

Erfahrungen mit IT-Kaizen im Projekt 2/2

- Während die ersten 3 Monate noch regelmäßig Verbesserungsvorschläge kamen, nahm es danach rapide ab; heute nach 11 Monaten faktisch keine Verbesserungsvorschläge mehr
  → wir hätten erneut einen Event/Aufruf, neue Kampagne starten sollen
- Qualitätszirkel mit Elan gestartet, Umsetzung der erarbeiteten Vorschläge aber im Projekttagesgeschäft untergegangen
  → explizit Zeit für Umsetzung einplanen!
- Tendenz zu Architektur- und Code-Diskussionen im Bezug auf IT-Kaizen schwer zu unterdrücken und dabei sehr unterschiedliche Vorstellungen (was mit der vorhandenen Codebasis erst alles passieren müsste, bevor es besser werden kann…)

Stimmen der Entwickler zu IT-Kaizen 1/5

- 12 von 13: **Die Kaizenvorstellung im Dezember 2005 fand ich gut.**
- 5 von 13: **Bei der Vorstellung hatte ich mir mehr davon versprochen als später eingetreten ist.**
- 10 von 13: **Ich finde generell die Kaizen-Idee gut.**
- 0 von 13: **Ich halte generell nichts von der Kaizen-Idee.**

Stimmen der Entwickler zu IT-Kaizen 2/5

- 9 von 13: **Unser Prozess hat sich seit Ende 2005 verbessert.**
- 4 von 13: **Die Verbesserung würde ich zumindest teilweise auch auf Kaizen zurückführen.**
- 4 von 13: **Die Verbesserung hat nur wenig mit Kaizen zu tun.**

- 8 von 13: **Ich denke selber unabhängig von Kaizenvorschlägen mehr über meine Arbeit nach.**

- 3 von 13: **Ich habe den Eindruck, dass ich anfangs mehr über Verbesserungen nachgedacht habe.**

Stimmen der Entwickler zu IT-Kaizen 3/5

- 11 von 13: **Prozessverbesserungen sind eher Teamangelegenheit.**
- 9 von 13: **Die Standardisierungen sollte man im Team vereinbaren, nicht für das ganze Projekt.**
- 5 von 13: **Die Standardisierungen sollten schon über das ganze Projekt erfolgen.**

Stimmen der Entwickler zu IT-Kaizen 4/5

- 7 von 13: **Die Kollegen halten sich nicht an die vereinbarten Standardisierungen.**
- 10 von 13: **Die umgesetzten Verbesserungsvorschläge waren mehrheitlich gut.**
- 3 von 13: **Die Verbesserungsvorschläge werden zu langsam umgesetzt.**
- 8 von 13: **Anfangs waren alle motiviert durch den Reiz des Neuen. Jetzt ist die Luft raus.**

Stimmen der Entwickler zu IT-Kaizen 5/5

- **0** von 13: **Ein finanzielles Belohnungssystem hätte zu mehr oder besseren Vorschlägen geführt.**
- 6 von 13: **Die Qualitätszirkel waren eine gute Idee.**
- **1** von 13: **Ich hatte den Eindruck, dass ich nicht genug Zeit hatte, um über Verbesserungen nachzudenken.**
- **2** von 13: **Ich habe mir nicht genug Zeit genommen, um über Verbesserungen nachzudenken.**

Unser Fazit

- agile Softwareentwicklung enthält einen Verbesserungsprozess
- IT-Kaizen ist kein Ersatz für Retrospektiven
- IT-Kaizen kann zusätzliche Impulse geben
- Für den Verbesserungsprozess/Umsetzungen muss Zeit gegeben werden
- Die Umsetzungen müssen sehr direkt erfolgen
- Projekthandbücher können auch in agilen Projekten nützen (wenn man es nicht übertreibt):
  - Als Orientierung gerade für neue Entwickler
  - Zum Nachschlagen für alle (vor allem für seltenere Vorgänge)
  - Zum Erinnern (warum machen wir etwas so? was haben wir uns gedacht?)

akquinet

Vielen Dank für die Aufmerksamkeit

Noch Fragen?

Schulung verlängerte Werkbank
agile Softwareentwicklung
Festpreisprojekte Coaching
RCP **Systemintegration** Eclipse
h3270 Hostintegration
Scrum Refactoring testgetriebene Entwicklung
Hibernate SAP-Netweaver **OpenSource**
Ajax JBoss/JEMS Groovy
it-agile eXtreme Programming

akquinet