



## **Kooperatives Testen**

Basis auch zur Testautomatisierung  
während der Softwareentwicklung

Dipl. Inform. Hans-Josef Eisenbach



## Der rote Faden

- Motivation zum Testen während der Softwareentwicklung
- Das Testkonzept „Kooperativen Testen“
- Die Anforderungen
  - abgeleitet aus dem Testkonzept
  - zur erfolgreichen Testautomatisierung
- Die Folgen und Benefits für den Prozess





## Historische Testentwicklung

- „Software ist wie eine Banane und reift beim Kunden“
  - ☹ *Folgekosten sind nicht abschätzbar*
  - ☹ *Imageverlust („Ist der Ruf erst ruiniert, lebt es sich ganz ungeniert“)*
- **Abnahmetest, Pilotierung durch den Kunden**
  - 😊 *Kunde entscheidet mit über Inbetriebnahme*
  - ☹ *Usability Prüfung erfolgt zu spät!*
  - ☹ *Fehler werden zu spät und durch den Kunden gefunden!*
- **Systemtest am Ende der Entwicklung**
  - 😊 *Entwickler testen systematisch die Software und können früher Fehler korrigieren*
  - ☹ *Testen erst am Ende der Entwicklung mit hohem Zeitaufwand und hohem Risiko*



## Ziele des strukturierten Testens


- Fehlerkostenreduktion 
- Senken des Prozessrisikos
- Steigerung der Akzeptanz beim Kunden
- Transparenz des Softwareentwicklungsprozesses
- gesunde Kosten/Nutzen Relation des Testens
- automatisches Testen 
- Motivation zum Testen in der Softwareentwicklung
- Steigerung des Fehlerbewusstseins in der Entwicklung
- ...

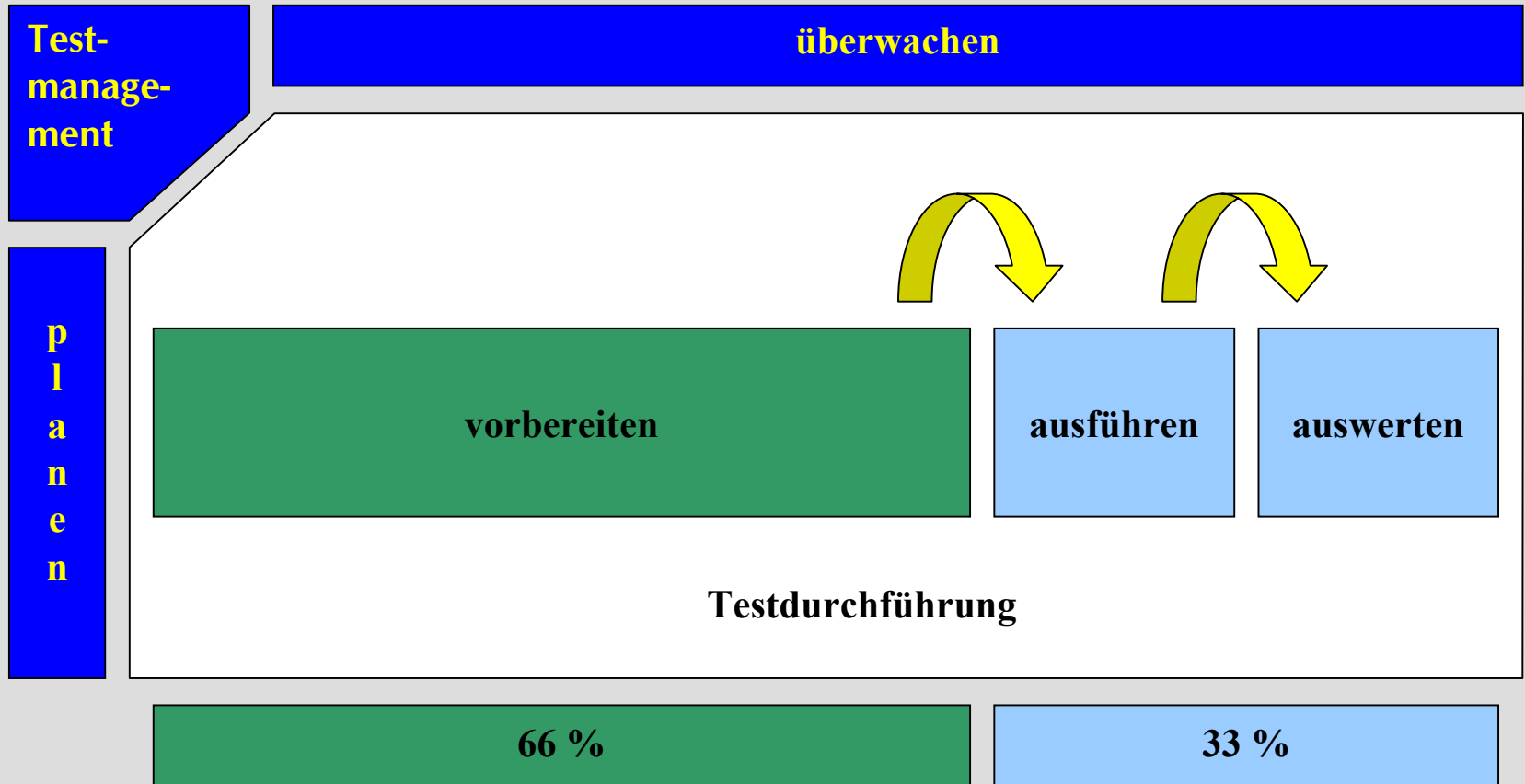




## Kosten der Fehlerbeseitigung im Verlauf der Systementwicklung

Phase	Kosten
Definition	1 Euro
Allgemeines Design	2 Euro
Detailliertes Design	5 Euro
Programmierung	10 Euro
Einheitentest	15 Euro
Integrationstest	22 Euro
Systemtest	50 Euro
Nach Auslieferung	>100 Euro

(Quelle: E. Dustin et al, Software automatisch testen, Springer) 



(Quelle: Dr. E. Wallmüller, Qualität und Informatik) ◀

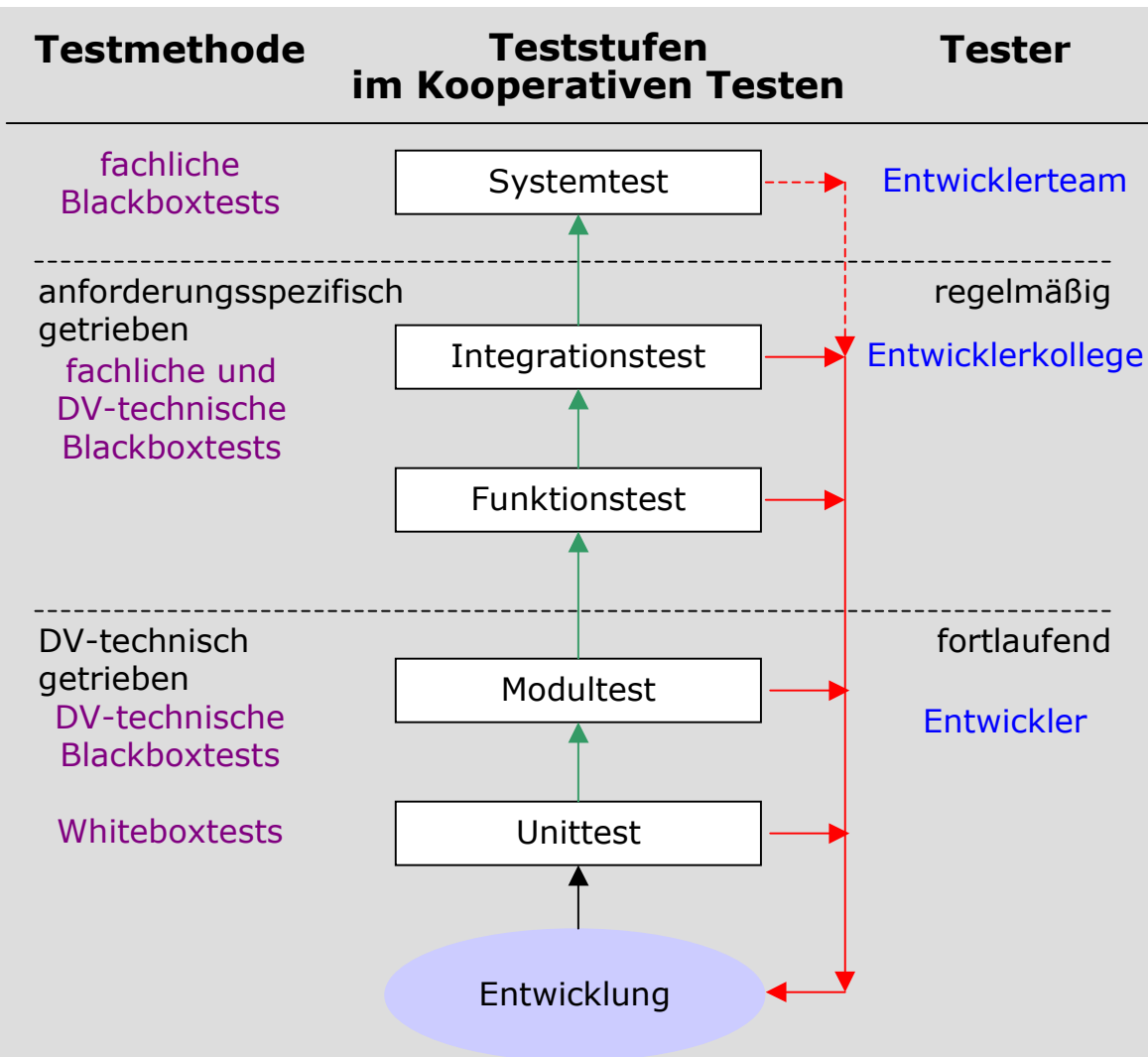


## Die Idee ist „Kooperatives Testen“

**intensives Testen während der Softwareentwicklung**

**jeder Softwareentwickler ist gleichzeitig Tester**

- > die Fehleranalyse kann schon durch den Tester erfolgen
- > der Tester erhält als Entwickler einen Überblick über das Softwareprodukt (Know-How-Transfer)
- > der *Tester* profitiert von seiner *Entwicklernerfahrung*
- > der *Entwickler* profitiert von seiner *Testerfahrung*





## Automatisierung des Funktions- und Integrationstests

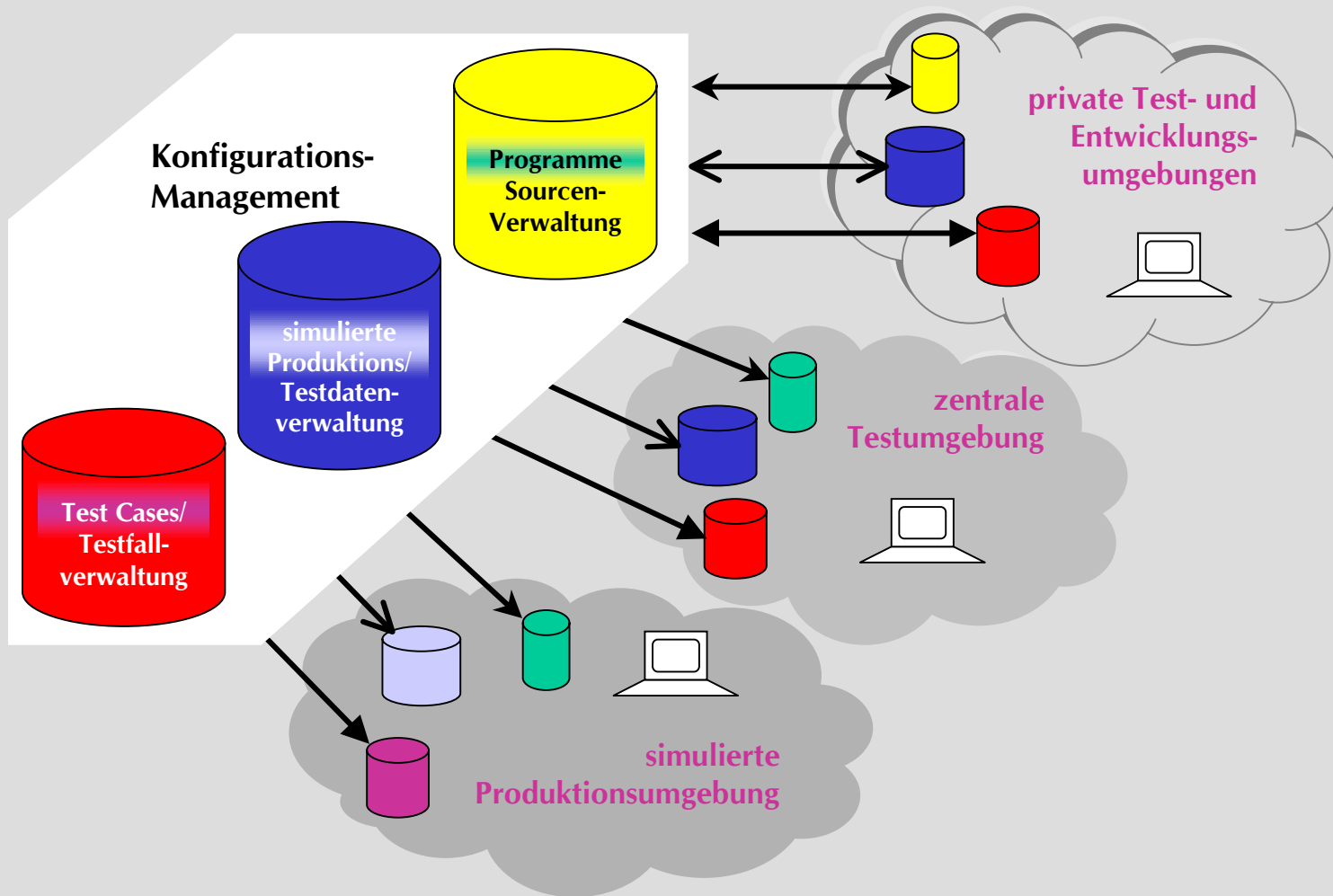
- Nightly Build
  - Erzeugend der zu testenden Produkte
- Nightly Test
  - Testen der im Nightly Build erzeugten Produkte
  - Testumfang ist auf eine Nacht beschränkt
  - Test wird jede Nacht wiederholt
- Weekly/Monthly Test
  - Testen der im Nightly Build erzeugten Produkte
  - Testumfang wird erweitert
  - Test wird jedes Wochenende bzw. einmal pro Monat wiederholt

- > Programmsourcen, Testfälle und Testdaten gemeinsam verwalten
- > Reproduzierbare Testumgebung



## reproduzierbare Testumgebung

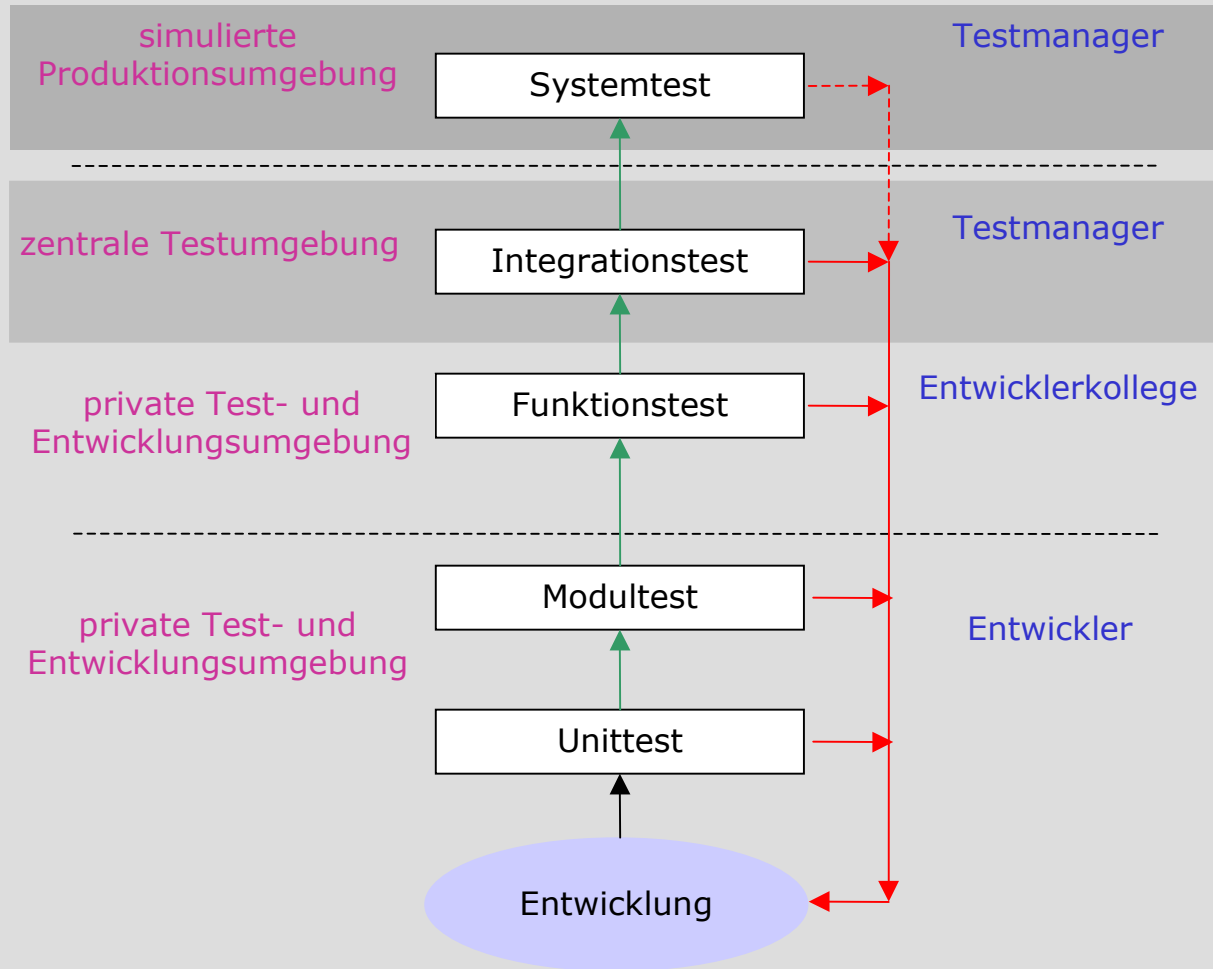
- > Jeder Test/Fehlerfall kann jederzeit wiederholt werden
- > Die Vorbereitung für Nachtests kann drastisch gesenkt werden
- > Das Testen in der Softwareentwicklung kann gezielt automatisiert werden
- > Die Software wird in einer simulierten Produktionsumgebung entwickelt





## Testumgebungen im Kooperativen Testen

## Umgebungsverantwortlicher





## Anforderungen an die Testumgebung

- Buildrechner
  - Betriebssystem
  - Prozessor
  - Compileroptionen
- Testrechner
  - Betriebssystem
  - Prozessor
  - Anzahl der Prozessoren
  - Netzwerkanbindung
- Cluster
  - Anzahl der Rechner
  - Klassifizierung der Rechner



## Anforderungen an Testwerkzeuge

- **Verwaltungswerkzeuge**
  - Sourcen, Testfälle (Subversion, Unix Open Source)
  - Fehler, Anforderungen (Trac, Open Source)
- **Testwerkzeuge in der Softwareentwicklung**
  - Compiler / Linker (Buildprozess, gcc, Open Source)
  - Speicherverbrauch (Rational purify ?)
  - Lastverhalten (Rational quantify ?)
  - Pfadabdeckung (Rational covarage ?)
- **Werkzeuge zur Unterstützung der Regressionstests**
  - Unittesting (cppunit, Open Source)
  - Capture und Replay Tools (x-Runner, mercury interactive)
  - Scripting, Simulationen (Test-Framework)



## Anforderungen an den Testlauf

- Jedes Modul / Produkt einzeln erzeugen
- Jedes Modul / Produkt alleine testen
  - Initialisierung
  - Testen (Testsuite)
  - Deinitialisieren
- Gesamtsystem erzeugen
- Gesamtsystem testen
  - System starten
  - System funktional testen
  - System herunterfahren

Funktionstest

Integrationstest



## Zusätzliche Anforderungen

- Testbarkeit sicherstellen
- Synchronisation des Testlaufs
- Wechselnde Testbedingungen berücksichtigen
  - Konfigurationen
  - Verwendete Testdaten
    - aus Referenzdaten
    - neu berechnete Daten
- Definition von Metriken
  - Bestimmung des Testaufwandes
  - Darstellung des Projektfortschrittes
  - Ermittlung der Komplexität
  - Definition geeigneter Testabbruchkriterien



## Jede Änderung hat Folgen

Das „Kooperative Testen“ in den Softwareentwicklungsprozess zu integrieren hat Folgen:

- für das Prozessmanagement
- für die Softwareentwicklung
- für das Konfigurationsmanagement
- für das Qualitätsmanagement



## Folgen für das Prozessmanagement

- Bereitstellung von fachlichen Musterprozessen
  - Bereitstellung von Testdaten
  - Iteratives Vorgehen
  - Definition kleiner überschaubarer (Teil-) Aufgaben
  - Jede Aufgabe hat einen Entwickler und einen Tester
  - Festlegung von Meilensteinen
  - Verifikation durch geeignete Teststufen
- 
- > ***Transparenz im Prozess***
  - > ***Risikoreduzierung im Projekt***
  - > ***Know - How Transfer im Team***



## Folgen für die Softwareentwicklung

- Definition von Testfällen
  - Erstellung von Testdaten
  - Einhalten von Qualitätsstandard bzgl. der Teststufen
  - Anwendung von Testautomaten
  - Verifikation durch Regressionstest
- 
- > ***Steigerung der Qualität durch frühes Testen***
  - > ***Verminderung der Fehlerkosten***
  - > ***Motivationssteigerung***



## Folgen für das Konfigurationsmanagement

- Möglichkeiten zur Pflege von Testfällen
  - Möglichkeiten zur Pflege von Testdaten
  - Überwachung von Qualitätsstandard (Teststufen, Richtlinien)
  - Bereitstellung von Informationen für Prozessverantwortliche
  - Bereitstellung der geforderten Testumgebungen
  - Bereitstellung der geforderten Entwicklungsumgebung
- 
- > ***Steigerung des Automatisierungsgrades***
  - > ***Basis zur Anwendung von Metriken***
  - > ***Steigerung der Reproduzierbarkeit des Vorgehens***



## Folgen für das Qualitätsmanagement

- Beratung zur Einführung
  - Treiber bei der Einführung
  - Controller bei der Nutzen/Kosten Analyse
  - Schulung der Mitarbeiter
- 
- > ***vorübergehende Mehrarbeit als Consultant***
  - > ***Steigerung der Qualität***



... und zum Schluss ...

noch Fragen oder Anregungen zum „Kooperativen Testen“?

***anrufen !!***

**Kontakt: EMPRISE Consulting Düsseldorf GmbH**

Hans-Josef Eisenbach

Dipl. Inform.

Cubus am Seestern

Niederkasseler Lohweg 18

40547 Düsseldorf

t\_ 0211 - 530 637 - 0

f\_ 0211 - 530 637 - 99

e\_ [hans-josef.eisenbach@emprise.de](mailto:hans-josef.eisenbach@emprise.de)

<http://www.emprise.de/ecd>