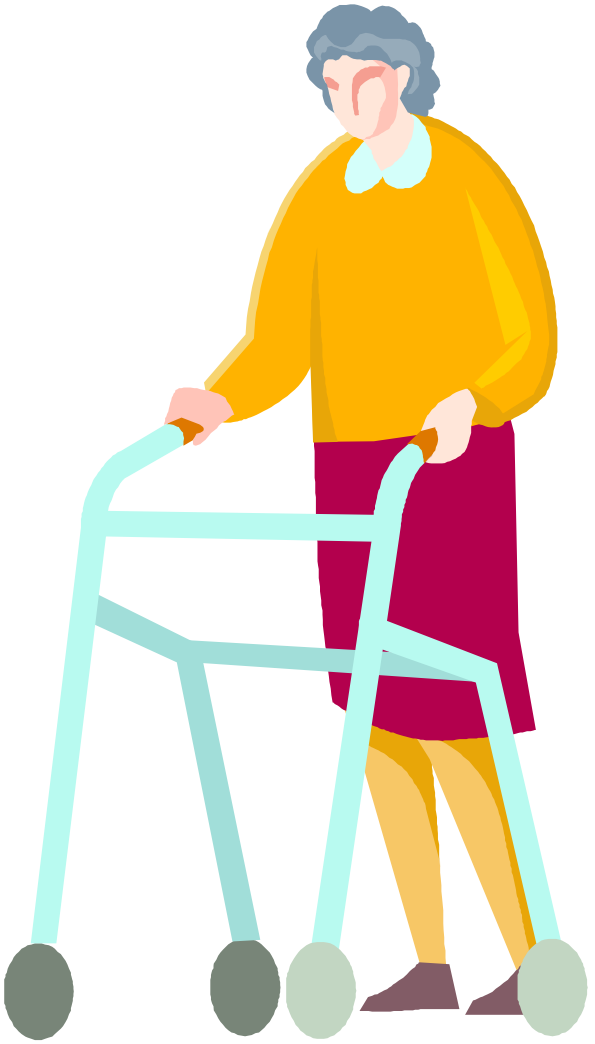
Anti-Aging für Embedded Software

Frank Listing
f.listing@microconsult.com

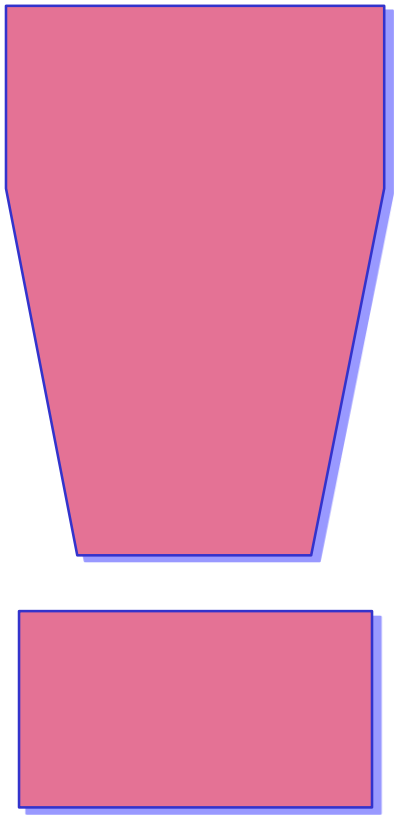
Wie soll Ihre Software nach ein paar Jahren Aussehen?

So?



Oder so?

Tipps für erfolgreiche Softwareprojekte



Vor der Geburt

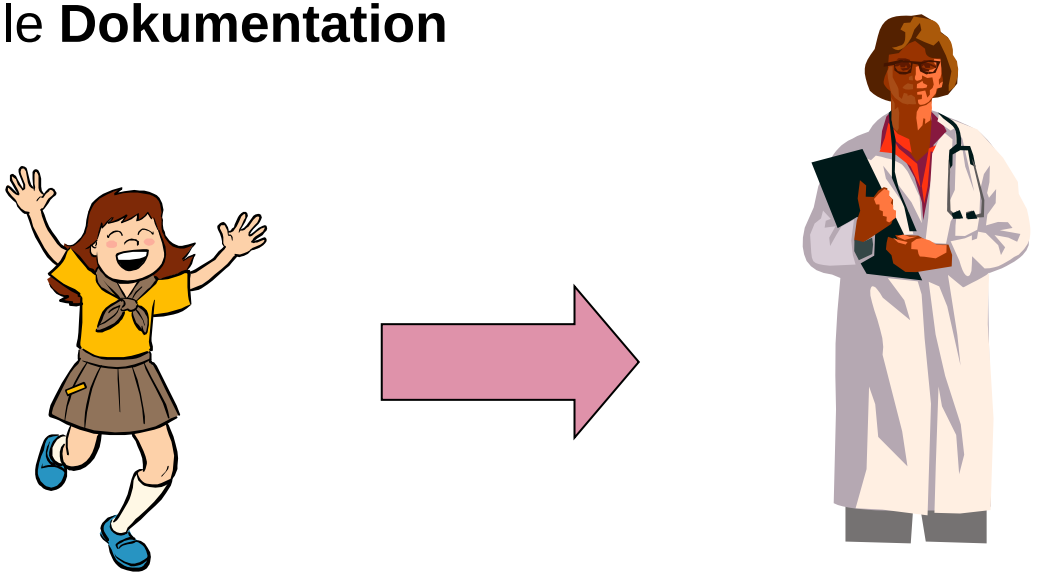
- **Ausreichend Zeit** in den **Projektstart** investieren.
- Aktiv mit dem Kunden zusammenarbeiten, um möglichst früh **vollständige Anforderungen** zu haben.

Etwas mehr Aufwand beim Projektstart bringt große Zeiteinsparungen in den späten Projektphasen.

Vorbild Mensch – Software mit Rückgrat

Das Skelett ist so gebaut, dass die Applikation mit den Anforderungen mitwächst.

- **solide Architektur**
- Komponenten mit **definierten Schnittstellen**
- gute **Strukturierung** des **Codes**
- sinnvolle **Dokumentation**

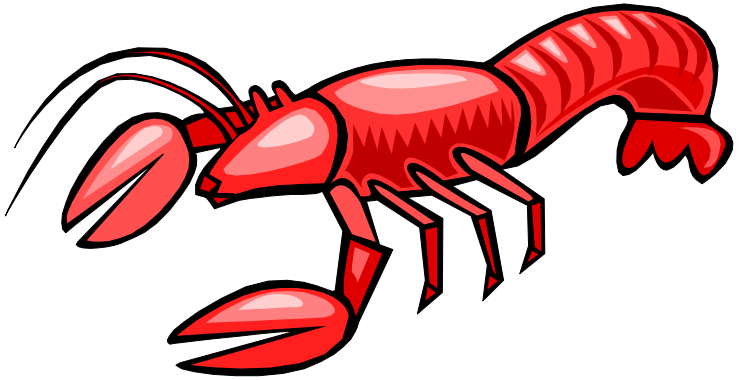


Ältere monolithische Projekte haben die Struktur eines Schalentiers.

Doch so ein Krebs ist hier ein schlechtes Vorbild:

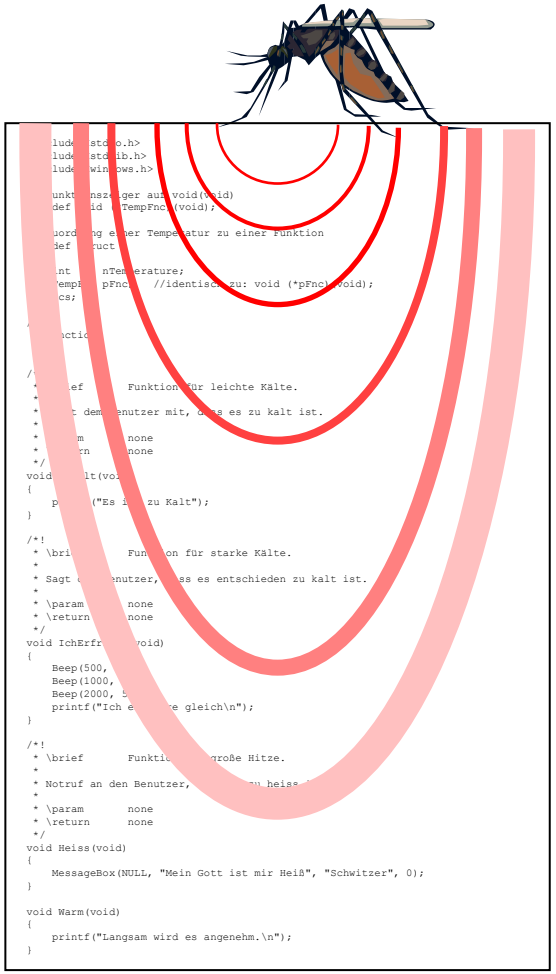
Wird der Körper zu groß, muss ein neues Gehäuse her.

Für die Software heißt es:
Bei Anforderungsänderungen muss der Code neu erstellt werden.

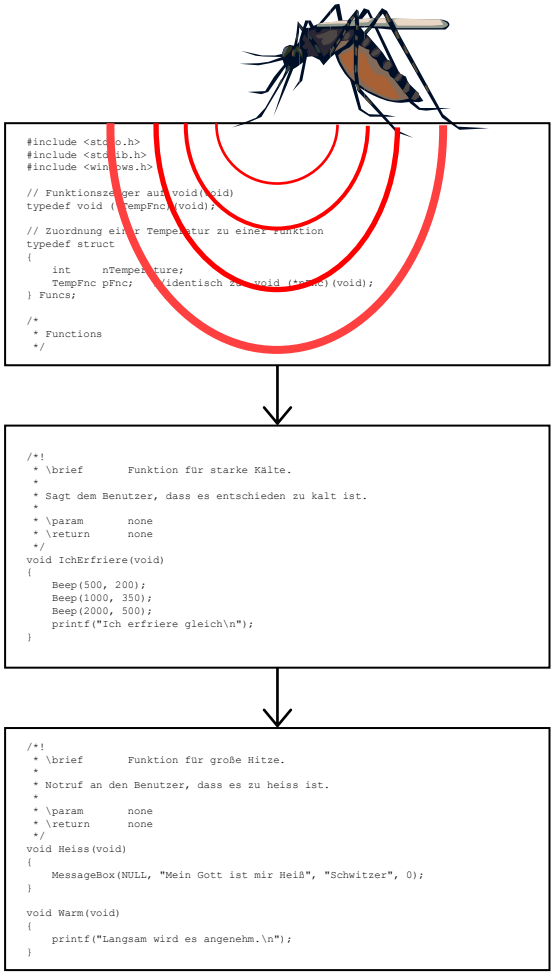


Gesund bleiben – Ansteckung vermeiden

Durch **abgeschlossene Komponenten/Module** wirken sich Fehler meist nur lokal aus und nicht in der gesamten Applikation.



```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

// Funktionszeiger auf void(void)
typedef void (*TempFnc)(void);

// Zuordnung einer Temperatur zu einer Funktion
typedef struct
{
    int     nTemperature;
    TempFnc pFnc;       //identisch zu: void (*pFnc)(void);
} Funcs;

/*
 * Functions
 */

/*!
 * \brief       Funktion für leichte Kälte.
 *
 * Sagt dem Benutzer mit, dass es zu kalt ist.
 *
 * \param       none
 * \return      none
 */
void Kalt(void)
{
    puts("Es ist zu Kalt");
}

/*!
 * \brief       Funktion für starke Kälte.
 *
 * Sagt dem Benutzer, dass es entschieden zu kalt ist.
 *
 * \param       none
 * \return      none
 */
void IchErfriere(void)
{
    Beep(500, 200);
    Beep(1000, 350);
    Beep(2000, 500);
    printf("Ich erfriere gleich\n");
}

/*!
 * \brief       Funktion für große Hitze.
 *
 * Notruf an den Benutzer, dass es zu heiss ist.
 *
 * \param       none
 * \return      none
 */
void Heiss(void)
{
    MessageBox(NULL, "Mein Gott ist mir Heiß", "Schwitzer", 0);
}

void Warm(void)
{
    printf("Langsam wird es angenehm.\n");
}
```



Gesund bleiben – Ausreichend Vitamine

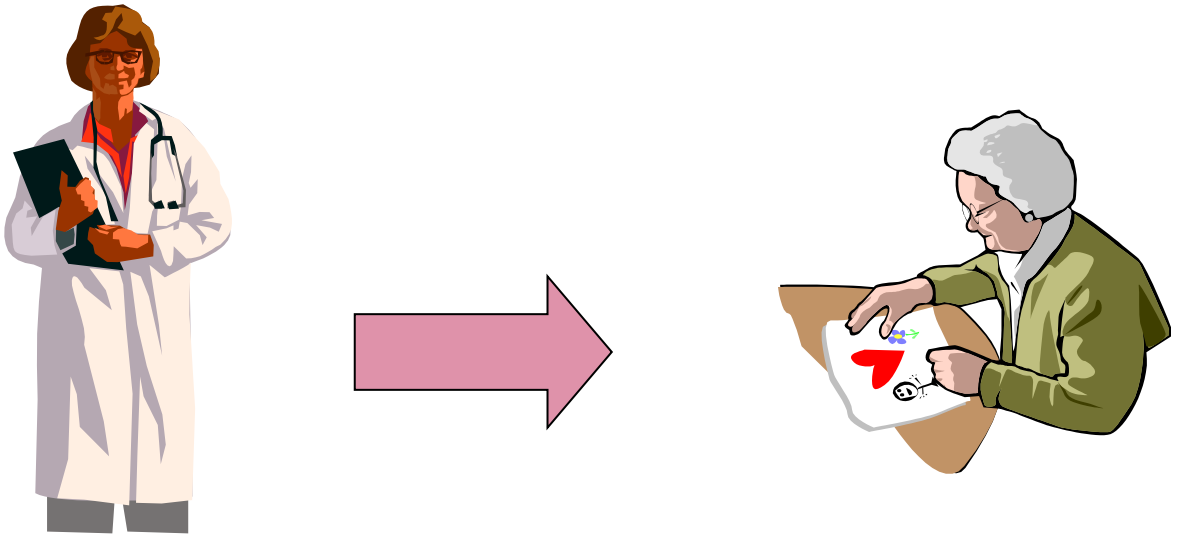
Sorgen Sie dafür, dass das **nötige Wissen** vorhanden ist

→ **Schulen Sie Ihre Mitarbeiter**

Gesund bleiben – Regelmäßige Gesundheitschecks

Durch kontinuierliche Pflege und Gesundheitschecks wird die Software lange gesund gehalten.

- **Einsatz** von **Codier-** und **Dokumentationsrichtlinien**
- **regelmäßige Prüfungen**, ob gegen Regeln (Architektur, Codierung) verstoßen wurde



Gesunder Lebenswandel

Durch den Einsatz von **Codier-Richtlinien** kann ein großer Schritt in Richtung gesunder Software getan werden.

- **Verbot kritischer Konstrukte**

  Senkung der Fehleranfälligkeit
- **einheitliche Strukturierung**

  Ordnung im "Kleinen" – Fehler werden schneller gefunden und können korrigiert werden.
- **einheitliche Namensgebung**

  einfachere Fehlersuche in "fremdem" Code

  schnellere Reviews
- **Dokumentationsrichtlinien**

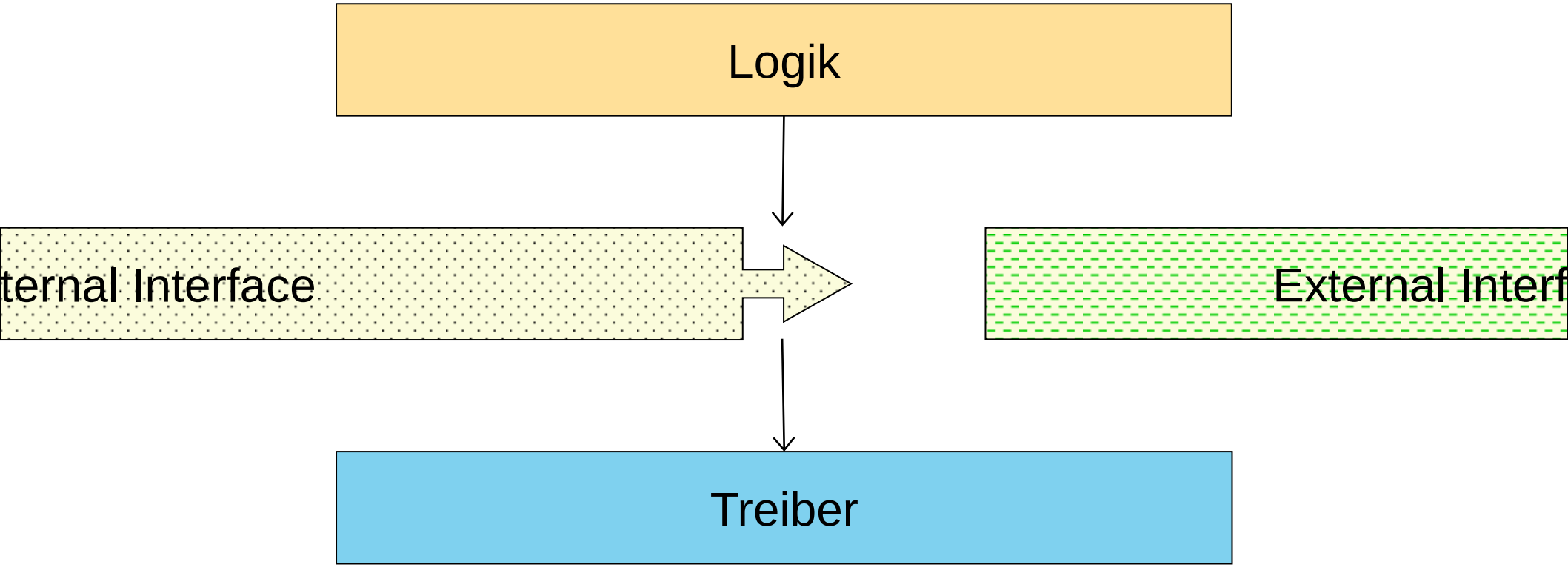
  schnellere Einarbeitung neuer Mitarbeiter
- **Einsatz von Dokumentations-Tools**

  aktuelle Dokumentation nach jedem Build

Operationen vereinfachen

Definierte Interfaces

- bewirken eine **einfache Austauschbarkeit der Komponenten** bzw.
- ermöglichen eine **Wiederverwendung** in Parallel- oder Folgeprojekten.



Gesund bleiben

- Eine Software ist gesund, wenn Körperbau (**Architektur**) und Organe (**Code**) gesund sind.
- **Regelmäßige Checks** erhalten die Gesundheit.
- Alle Mitarbeiter müssen auch mit den **Richtlinien** vertraut sein, um die benötigte **SW-Qualität** zu gewährleisten.

Zu Risiken und Nebenwirkungen,

vor allem zu Hinweisen, wie man diese beseitigt,

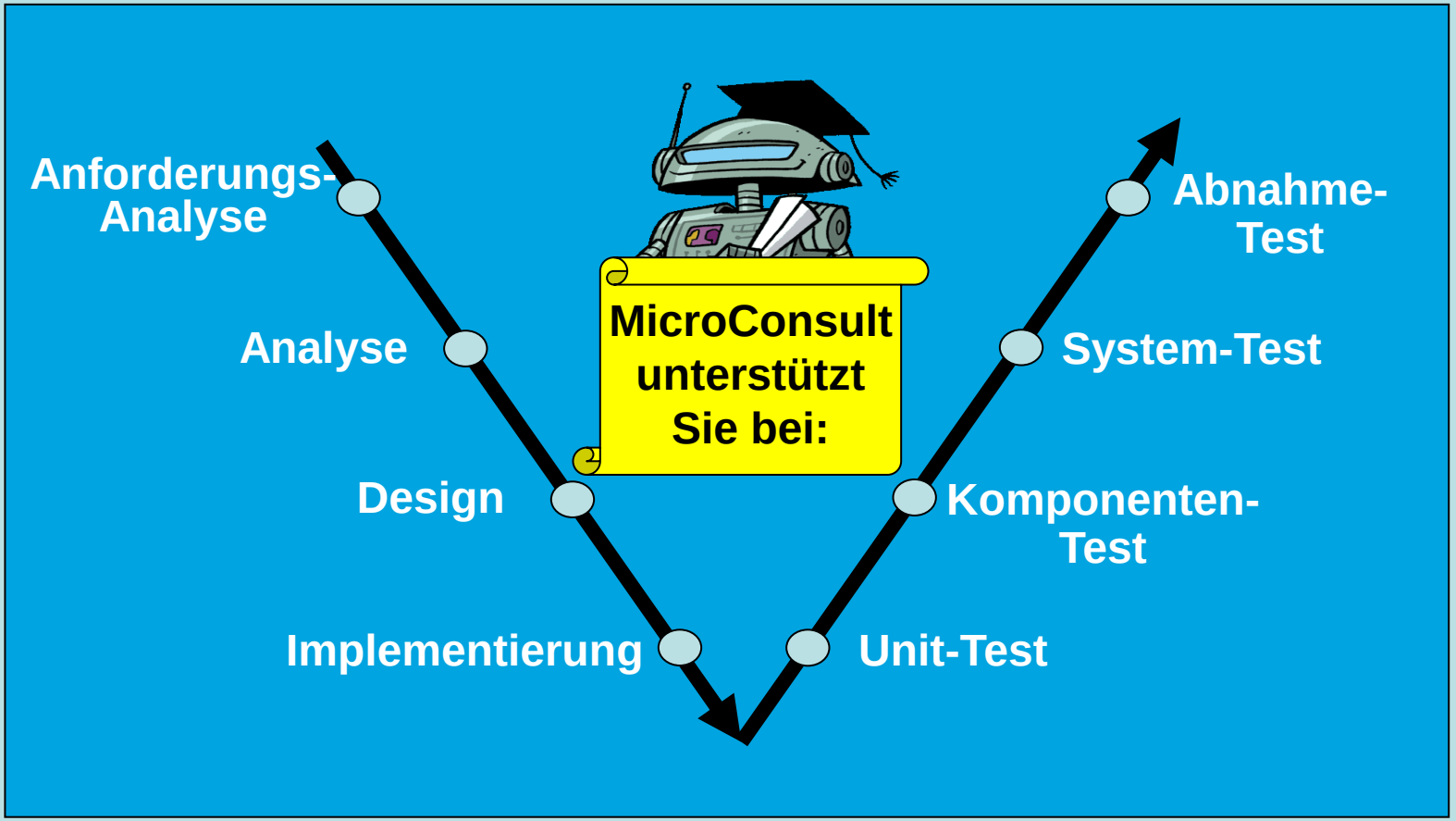
fragen Sie die Kollegen

von

MICRO CONSULT

MICRO CONSULT

Beratung, Training, Workshops, Coaching, Consulting



HW-/SW-Technologien, Tools, Methoden, Prozess, Team

Vortrag per E-Mail anfordern: info@microconsult.de
Betreff: „Vortrag Embedded World 2013“