



*Bernhard Nett:*

# *Partizipative Technikentwicklung*

gefördert vom MKW NRW  
mit Unterstützung des FGW

Ministerium für  
Kultur und Wissenschaft  
des Landes Nordrhein-Westfalen



Forschungsinstitut  
für gesellschaftliche  
Weiterentwicklung



ARWISO e.V. • Karl-Carstens-Str.1 • 52146 Würselen  
[www.arwiso.de](http://www.arwiso.de)

# Digitaltechnik – etwas Besonderes?

- Technik: Naturobjekte, zweckmäßig bearbeitet + kombiniert
  - 100% zweckmäßig? Im Nachhinein ist man oft schlauer ...
  - Zwecke bestimmen die Technik – aber auch umgekehrt!
  - Technik selbst kontextabhängig, muss als solche erkannt werden
  - Formen und Verstehbarkeit von Technik kulturell, veränderlich
- Digitaltechnik
  - Doppelprodukt: Hardware + Software!
  - Software: „unsichtbare“ Code-Folge
  - Zwecke von Software nicht beGREIFbar
- Der Datenbegriff macht's nicht einfacher
  - Daten durch Hardware verarbeitet
  - Verarbeitung gemäß Programmen
  - Programme: ebenfalls Daten
  - Daten gewissermaßen Objekt und „Subjekt“ der Daten-Verarbeitung



Steinzeitlicher Steinhammer

# Software als Produkt

- Neuartige Ökonomie
  - Ein Computer (Hardware) – verschiedenste Anwendungen (SW)
  - Hardware kann zu Software gemacht werden. Und umgekehrt.
  - Technik für bestimmten Zweck kann verschieden „aussehen“, für verschiedene Zwecke gleich
  - Bei Software ist nicht der Computer, sondern die Anwendung das Produkt
  - Intuitive Bedienung ist Teil der Entwicklungsaufgabe (da Software nicht per se intuitiv ist)
  - SW-Produkte: der Prototyp sehr teuer, Kopien praktisch gratis
- „Alles ist Data“ – wirklich?
  - Daten können neue Daten generieren, Programme neue Programme (> Künstliche Intelligenz)
  - Jeder kann programmieren – wenn er/sie es kann! (Keine große Produktionstechnik nötig)
  - Aber: Vieles ist nicht in Datenform vorhanden – müsste erst dahin „übersetzt“ werden
- Wenn ein Programm offen, komponentenbasiert und objektorientiert programmiert ist, ...
  - können Datenstrukturen darin noch nachträglich verändert („vererbt“...) werden
  - kann seine Entwicklung einfach anhand der Nutzerschnittstelle erfolgen (vgl. Folie 8)
  - (Doch auch bei andern Programmen können eigene Skizzen zur Nutzerschnittstelle helfen zu verstehen - und den Entwicklern zu kommunizieren - was man eigentlich haben will.)

# Grundproblem der Technikentwicklung: „Symmetrie der Ignoranz“

- Kunden / Nutzer
  - wissen oft nicht, was technisch möglich ist
  - Kennen Probleme bei sich sehr gut, die gelöst werden sollten
  - verstehen Implikationen ihrer Probleme oft erst im Nachhinein
- Entwickler
  - wissen, was technisch möglich ist
  - verstehen den Nutzungskontext der Kunden oft nicht
  - drücken sich für Kunden oft unverständlich aus
- Verständnisprobleme
  - zwischen Auftraggebern und Auftragnehmern
  - zwischen verschiedenen Rollen bei Auftraggeber
  - führen oft zur mangelhaftem Verständnis / schlechtem Design

# Entwickler-Techniken: Szenarien

## Konzept

- Ein Szenario beschreibt mögliche zukünftige Situationen textuell. Folgen:
  - man kann sich so zukünftige Situationen vorstellen
  - Beteiligte können Einwände (z.B. zu Implikationen) machen
  - Man findet gemeinsam die beste Lösung
- „Übersetzung“ in techn. Spezifikationen erst nach dem Diskussionsprozess!
  - Beteiligte können sich in der für sie üblichen / verständlichen Art einbringen
  - Wenn IT-Spezialist schon einbezogen ist, darf er/sie nicht dominieren
  - Szenarien mildern die „Symmetrie der Ignoranz“
- Anreicherung des Szenarios
  - Ein Szenario wird zur Diskussion gestellt
    - Auf Teamtreffen, in einem Forum, per Aushang, ...
    - Kollegen können erkennen, was es für sie bedeuten würde
    - Sie können Szenario ergänzen, annotieren, eigene Beiträge löschen
  - Szenario kann analysiert werden, BEVOR Entscheidungen getroffen werden
  - Rahmenbedingungen können (müssen!) ausgehandelt werden
- Umsetzungsgrad: hilft prüfen, wie viel Gewünschtes verwirklicht wurde

# Entwickler-Techniken: Szenarien (fiktives) Beispiel

## **(Anfangs-) Szenario:**

*„Erwin erhält die Freigabe für seinen nächsten Auftrag auf seinem Handy. Er geht in Halle 2 zur 5-Achs-Fräse: da er dort schon sein Foto auf dem Monitor sieht, weiß er, dass er direkt loslegen kann: er hält das Display seines Handies mit dem zugemailten Barcode an den Scanner und lädt so die von der AV geprüften und komplettierten Fertigungsdaten aus der Datenbank in die Maschine.“*

## **Ergänzung vom Monteur Werner:**

*„Dabei wird die Benachrichtigung, dass die Teilelisten und vorbereiteten Detailpläne geladen werden können, auf das Handy von Werner geschickt, zusammen mit dem zugehörigen Zugangscode ...“*

## **Annotation des Arbeitsvorbereiters Klaus an Werners Ergänzung:**

*„Das klappt so nicht, weil zum Zeitpunkt der Fertigung die Teilelisten für die Baugruppe noch nicht geprüft sein müssen“*

## **Monteur Werner löscht seine Ergänzung und schreibt:**

*„Dabei wird eine Benachrichtigung über die Fertigung auf das Handy von Werner geschickt“*

# Entwickler-Techniken: Mock-Ups

## Was sind Mock-Ups?

- Mock-Ups simulieren eine Mensch-Maschine-Interaktion nur auf Papier
- Später können Nutzer simulieren, was der Computer tun soll
- Mock-Ups können auch Mehrbenutzeranwendungen simulieren
- Die Papier-Simulation erlaubt Nutzerbeteiligung an Technikentwicklung
- Sie erleichtert Prüfungen der Designannahmen
- Später reagiert ja der Computer: beim Mock-Up der Entwickler
- Abfolge von Mock-Ups gemäß Storyboard
- Storyboard enthält mögliche / typische Abläufe
- ... erlaubt gemeinsames Durchdenken möglicher Probleme und Alternativen
- ... kann bei Entwicklung des Systems helfen
- Mock-Ups verringern die Symmetrie der Ignoranz
- Mock-Ups sparen Kosten: Entwickler erkennen, was genau gewünscht wird
- Mock-Ups können nicht nur Programme simulieren, sondern auch Konfiguration

# Entwickler-Techniken: Mock-Ups

## Ein Beispiel

1. Der Entwickler hat ein Ringbuch mit Mock-Ups
2. Der Entwickler legt dem späteren Nutzer ein Mock-Up vor  
(Zum Einstieg in der Regel die Startseite - als bloßes Papierdokument!)
3. Das Mock-Up zeigt die Nutzerschnittstelle (z.B. Angaben und eine Eingabeaufforderung auf einer Touchscreen) auf Papier
4. Der Mitarbeiter liest die Eingabe-Abfrage auf dem (Papier-) Display ab
5. Der Mitarbeiter simuliert die abgefragte Eingabe von Zahlen und die Bestätigung auf dem Touchscreen (alles nur auf dem Papier)
6. Der Entwickler notiert auf eigenem Papier die Eingaben des Testnutzers
7. Der Entwickler blättert anhand seines Storyboards zu der Seite, die dann erscheinen würde

Weiter mit 1.



# Entwicklungs-Leitbilder

## Usability / End-User Development

- Usability

- z.B. Nutzerfreundlichkeit (z.B. Anwendung lokalen Vokabulars)
- z.B. Lernförderlichkeit (z.B. durch Historienfunktionen)
  - Wie hab ich das beim letzten Mal gemacht?
  - Kollektiv: wie hat der Kollege etwas gemacht?
- z.B. Fehlertoleranz (z.B. durch „UNDO“ / „zurück“)
  - den letzten Schritt ungeschehen machen

- EUD

- Entwicklung / Konfiguration durch Nutzer selbst
  - Design-Freiräume in Anwendungen einbauen
- Nutzerautonomie fördern
  - Kommunikationskanäle zu Lösungsexperten statt bloßer Fehlermeldung
  - „Hilfe“-Funktionalitäten an Systemzustände gekoppelt

# Kontinuierlicher Verbesserungsprozess (KVP):

Beschrieben in der ISO 9001 / Als Standardprozess der Softwarebranche

Blau: KVP nach ISO 9001

Grün: Softwarebranche



Für erfolgreiche Digitalisierungs-Strategie wichtig

In KMU mitunter zu wenig mit Leben gefüllt

**Evolutionäre / Iterative** (schrittweise) Weiterentwicklung von Software

# Schrittweise Technikentwicklung

- Schrittweise = iterative = evolutionäre Entwicklung
- weniger Risiko!
  - Nutzerbeteiligung hilft
  - Lernen aus der eigenen Praxis!
  - Pilotierung in begrenztem Versuchsbereich möglich
- Kein Zufall, dass die IT-Branche evolutionär arbeitet!
- Gut verbindbar mit Kontinuierlichem Verbesserungs-Prozess (KVP)
  - KVP in der ISO – 9001 gefordert
  - Vergleichbar: Organisationsentwicklung / Lernende Organisation / Organisationales Lernen
- „Innovation von unten“
  - braucht Ansprechpartner / Rahmen
  - ... kann passgenaue, gebrauchstaugliche Lösungen fördern
- Betriebliche Innovationskultur