

# **Datenbank-gestütztes Prüfsystem mit LabVIEW für ein Miele-Prüffeld**

**Autor: F. Sundmacher, Miele & Cie GmbH & Co., Werk Lehrte**

**Co-Autor: P. Schwarz, A.M.S. Software GmbH, Quickborn**

# Themen

- Die Firma Miele und das Werk Lehrte
- Anforderungen und Konzept
- Übersicht DPS
- Schnittstellen zu externen EDV-Systemen
- Aufbau der Prüfsoftware
- Ablauf der Prüfung
- Fehlermanagement



- Miele 1899 von Carl Miele und Reinhard Zinkann gegründet
- ca. 15.000 Mitarbeiter in der Miele-Gruppe
- Die Miele Werke

Gütersloh (Stammwerk)

Bielefeld

Euskirchen

Warendorf

Oelde

Bünde

Arnsberg

Bürmoos in Österreich

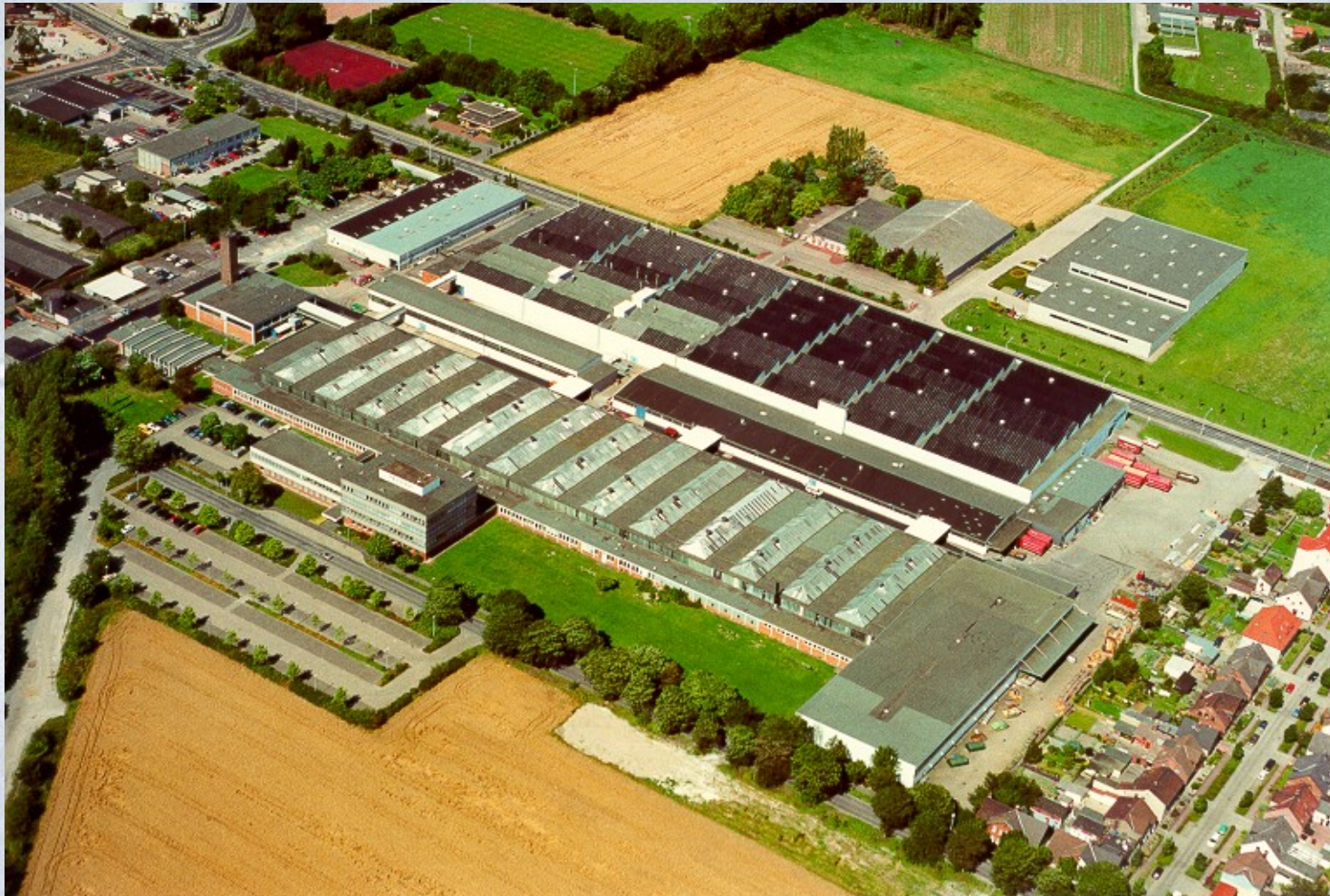
Lehrte



# Miele-Werk Lehrte

Produktion von Gewerbegeräten und  
Haushaltsbügelmaschinen.

**Miele**  
PROFESSIONAL





# Produkte Werk Lehrte

**Miele**  
PROFESSIONAL

## Waschen

Füllgewicht

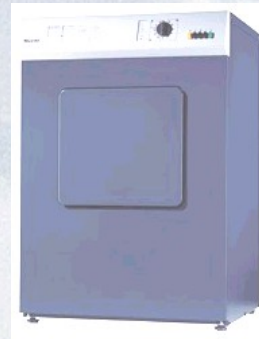
7,5 – 32 kg



## Trocknen

Füllgewicht

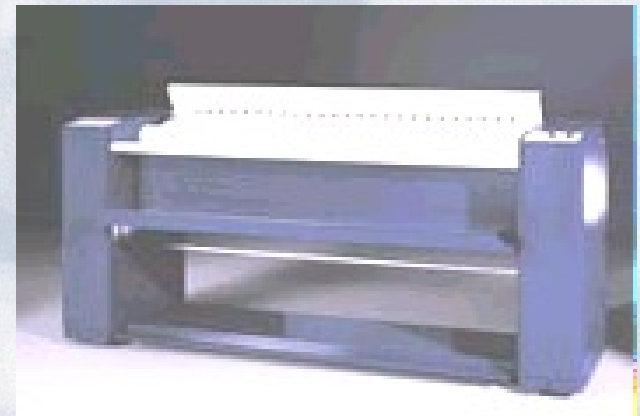
7,5 – 30 kg



## Mangeln

Walzendurchmesser

16 –88 mm



## Prüfsoftware

Prüffeld

Prüfstation 1

Prüfstation 2

Messtechnik

Prüfstation 3



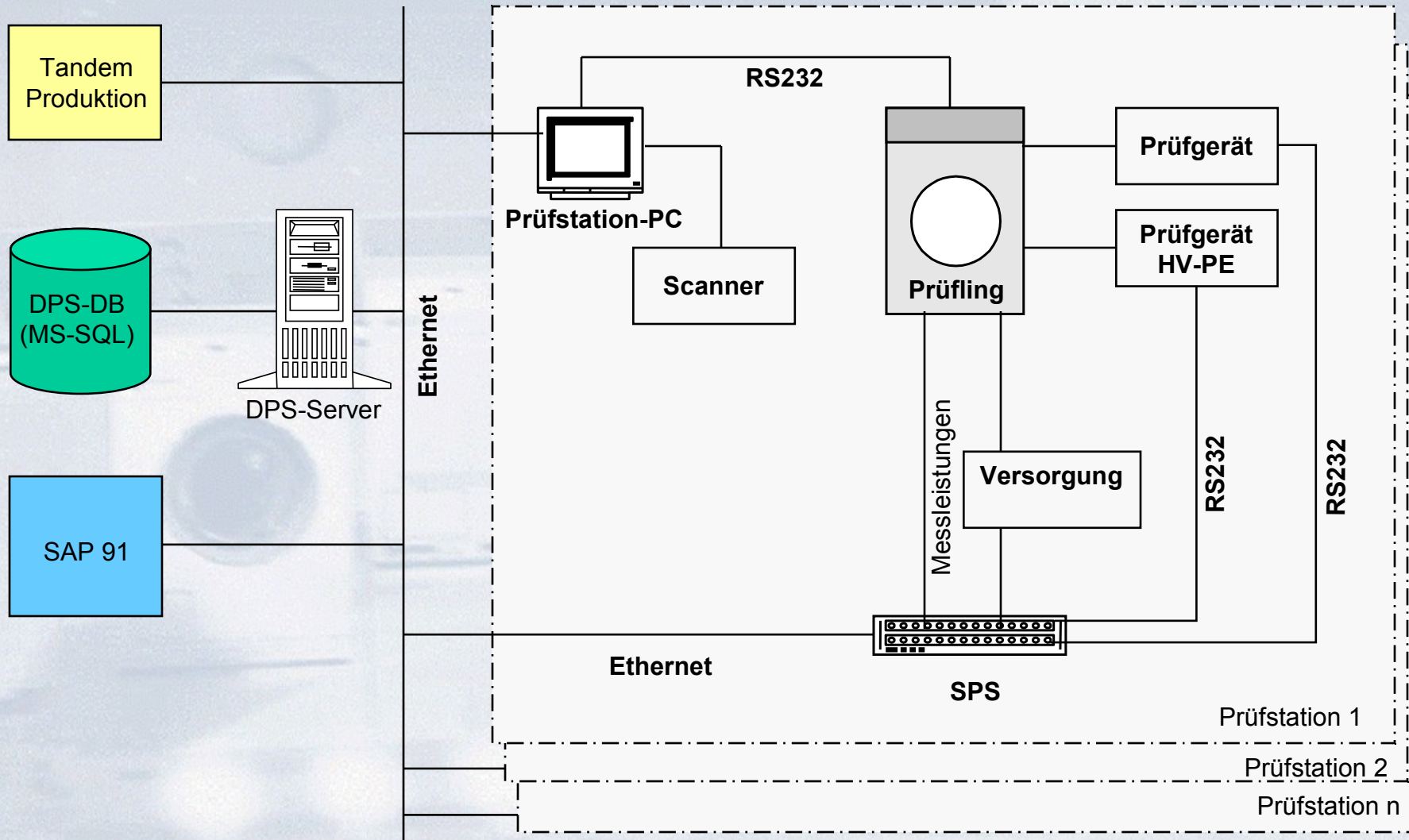
Arbeits-  
karte

DPS-DB  
(MS-SQL)

Prüfung

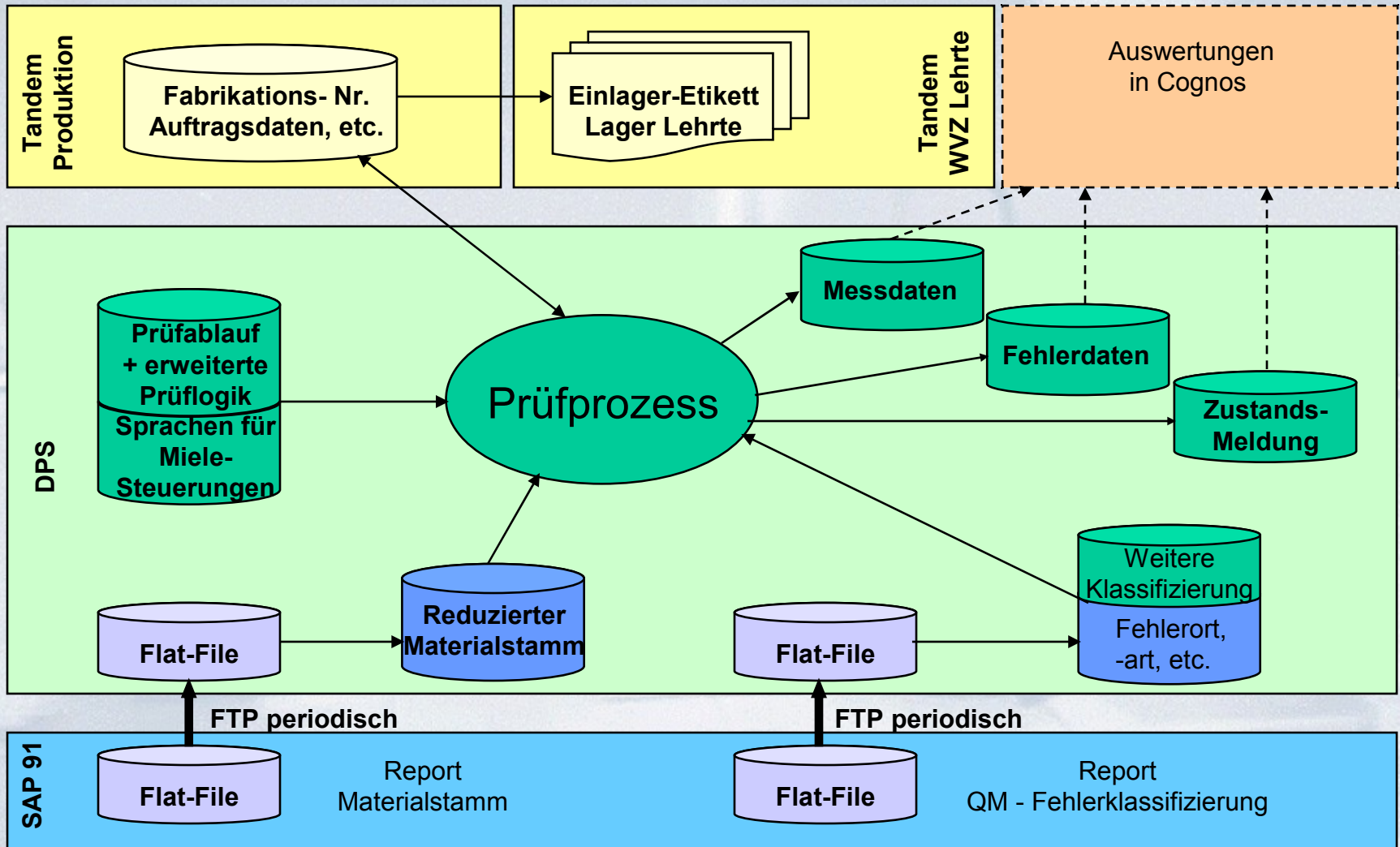
Fehler-  
management

# Übersicht DPS – Datenbank-gestütztes Prüfsystem



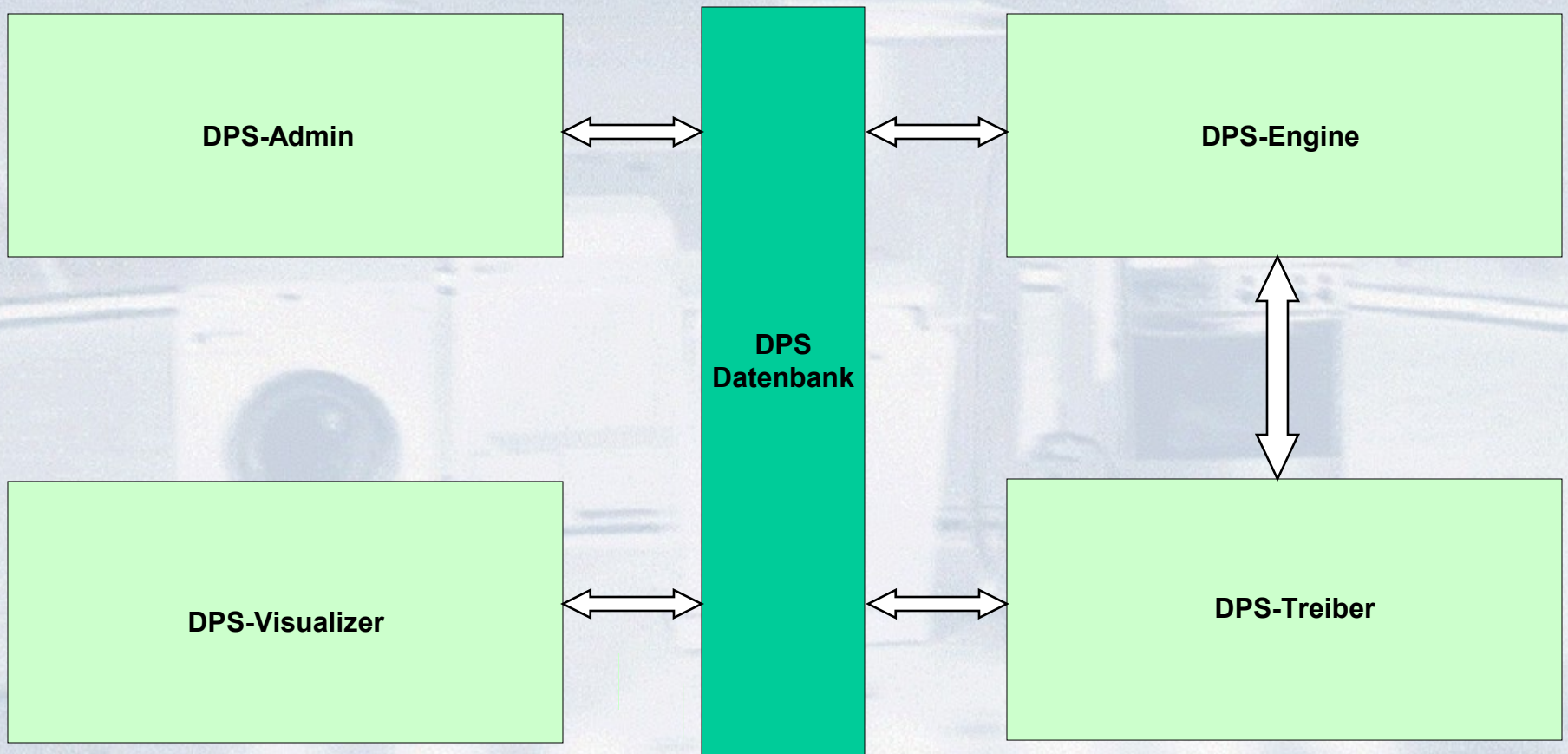


# Schnittstellen der EDV - Systeme

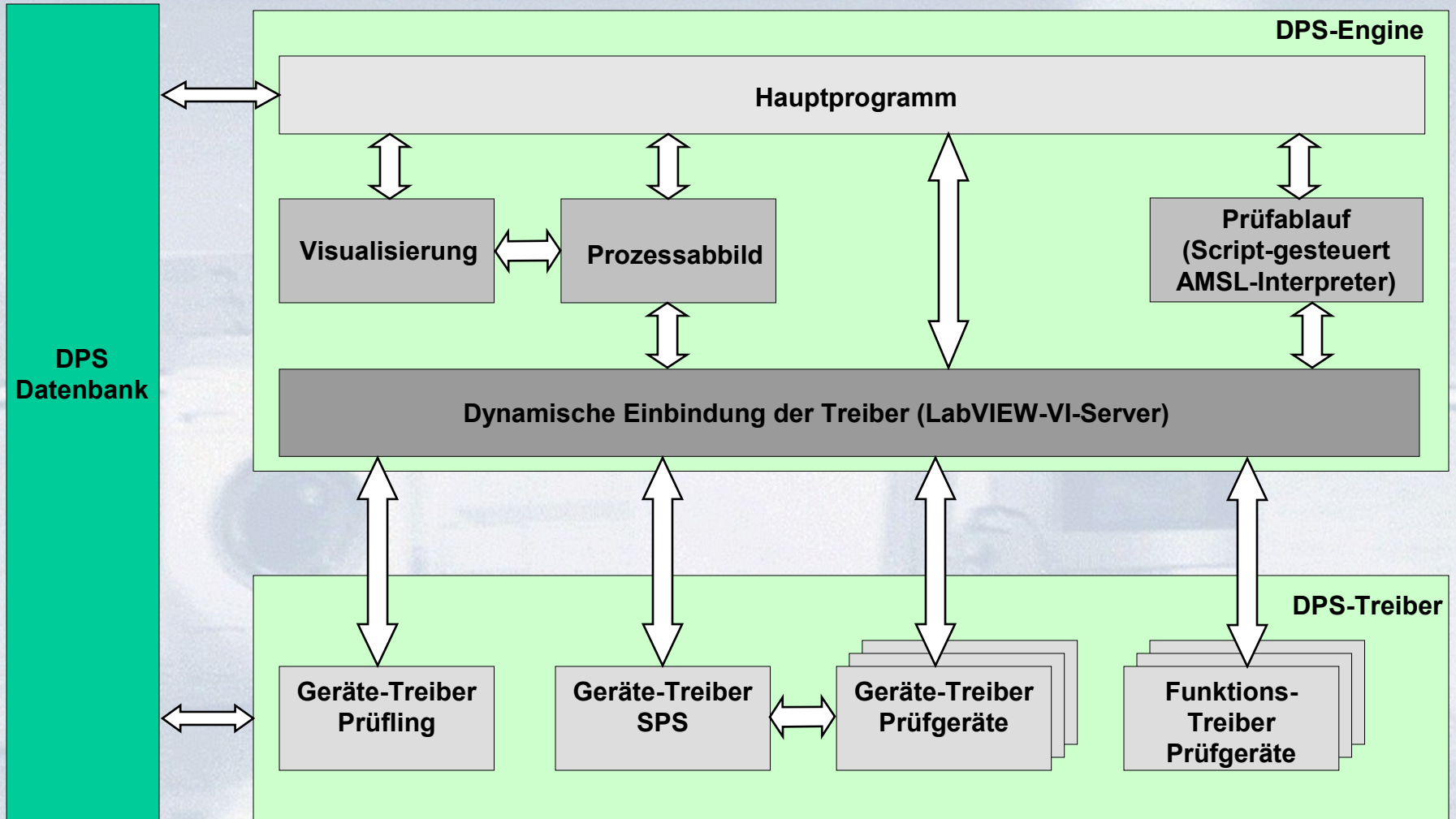




# Aufbau der Prüfsoftware

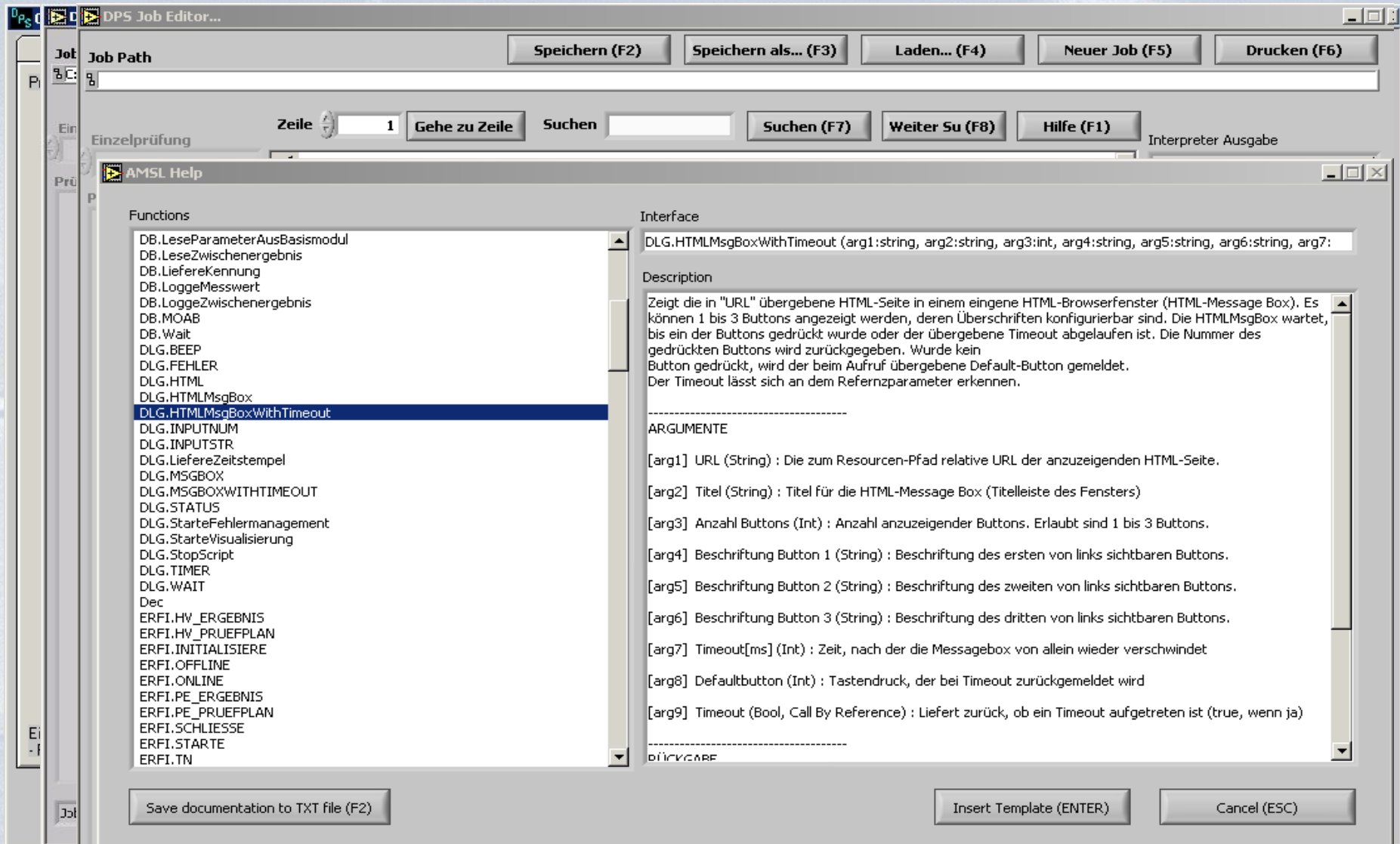


# Aufbau der Prüfsoftware





# Parametrierung und Prüfablauerstellung



The screenshot shows the 'DPS Job Editor' application window. At the top, there is a 'Job Path' field and several buttons: 'Speichern (F2)', 'Speichern als... (F3)', 'Laden... (F4)', 'Neuer Job (F5)', and 'Drucken (F6)'. Below this is a 'Zeile' (line) field set to '1' and a 'Suchen' (search) field. Further down, there are buttons for 'Suchen (F7)', 'Weiter Su (F8)', and 'Hilfe (F1)'. The main area is divided into two panes: 'Functions' on the left and 'Interface' on the right. The 'Functions' pane lists various functions, with 'DLG.HTMLMsgBoxWithTimeout' selected. The 'Interface' pane shows the signature 'DLG.HTMLMsgBoxWithTimeout (arg1:string, arg2:string, arg3:int, arg4:string, arg5:string, arg6:string, arg7:)', a description in German, and a list of arguments (ARGUMENTE) from [arg1] to [arg9]. At the bottom, there are buttons for 'Save documentation to TXT file (F2)', 'Insert Template (ENTER)', and 'Cancel (ESC)'.

**Functions**

- DB.LeseParameterAusBasismodul
- DB.LeseZwischenergebnis
- DB.LiefereKennung
- DB.LoggeMesswert
- DB.LoggeZwischenergebnis
- DB.MOAB
- DB.Wait
- DLG.BEEP
- DLG.FEHLER
- DLG.HTML
- DLG.HTMLMsgBox
- DLG.HTMLMsgBoxWithTimeout**
- DLG.INPUTNUM
- DLG.INPUTSTR
- DLG.LiefereZeitstempel
- DLG.MSGBOX
- DLG.MSGBOXWITHTIMEOUT
- DLG.STATUS
- DLG.StarteFehlermanagement
- DLG.StarteVisualisierung
- DLG.StopScript
- DLG.TIMER
- DLG.WAIT
- Dec
- ERFI.HV\_ERGEBNIS
- ERFI.HV\_PRUEFPLAN
- ERFI.INITIALISIERE
- ERFI.OFFLINE
- ERFI.ONLINE
- ERFI.PE\_ERGEBNIS
- ERFI.PE\_PRUEFPLAN
- ERFI.SCHLIESSE
- ERFI.STARTE
- ERFI.TN

**Interface**

DLG.HTMLMsgBoxWithTimeout (arg1:string, arg2:string, arg3:int, arg4:string, arg5:string, arg6:string, arg7:)

**Description**

Zeigt die in "URL" übergebene HTML-Seite in einem eingetragenen HTML-Browserfenster (HTML-Message Box). Es können 1 bis 3 Buttons angezeigt werden, deren Überschriften konfigurierbar sind. Die HTMLMsgBox wartet, bis ein Button gedrückt wurde oder der übergebene Timeout abgelaufen ist. Die Nummer des gedrückten Buttons wird zurückgegeben. Wurde kein Button gedrückt, wird der beim Aufruf übergebene Default-Button gemeldet. Der Timeout lässt sich an dem Referenzparameter erkennen.

-----

**ARGUMENTE**

- [arg1] URL (String) : Die zum Ressourcen-Pfad relative URL der anzuzeigenden HTML-Seite.
- [arg2] Titel (String) : Titel für die HTML-Message Box (Titelleiste des Fensters)
- [arg3] Anzahl Buttons (Int) : Anzahl anzuzeigender Buttons. Erlaubt sind 1 bis 3 Buttons.
- [arg4] Beschriftung Button 1 (String) : Beschriftung des ersten von links sichtbaren Buttons.
- [arg5] Beschriftung Button 2 (String) : Beschriftung des zweiten von links sichtbaren Buttons.
- [arg6] Beschriftung Button 3 (String) : Beschriftung des dritten von links sichtbaren Buttons.
- [arg7] Timeout[ms] (Int) : Zeit, nach der die Messagebox von allein wieder verschwindet
- [arg8] Defaultbutton (Int) : Tastendruck, der bei Timeout zurückgemeldet wird
- [arg9] Timeout (Bool, Call By Reference) : Liefert zurück, ob ein Timeout aufgetreten ist (true, wenn ja)

-----

**RÜCKGABE**

# Durchführung einer Prüfung

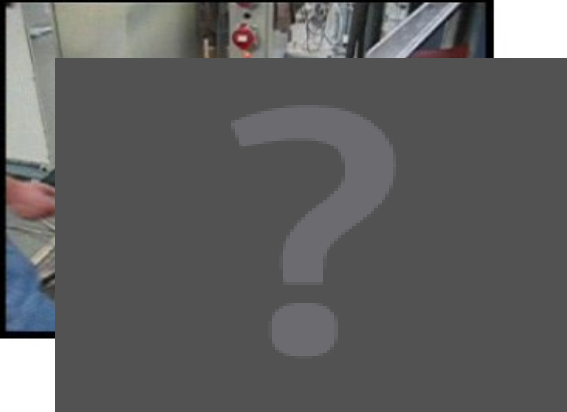
DPS-Prüfplatz: Funktion MG01-MG07 Erste

F	F	Fab.Nr.	90	Prüfer	Sund	19.02.2004, 12:51:20 : Einzelprüfung Probe Al memo 001 gestartet.
B	B	Bez.	TestAMS01	Datum	19.02.2004	19.02.2004, 12:51:20 : Modus Prüfung gestartet.
M	M	Mat.Nr.	4711	Uhrzeit	12:55.39	19.02.2004, 12:51:19 : Fertig.

19.02.2004, 12:51:19 : Gerät "SPS" - Konfiguration wird geprüft, Treiber = "SPS.vi" (8 von 8)-OK.  
19.02.2004, 12:51:19 : Gerät "Profi Licht" - Konfiguration wird geprüft, Treiber = "Profi1Intf.vi" (7 von 8)-OK.  
19.02.2004, 12:51:19 : Gerät "PROFI2INTE" - Konfiguration wird geprüft, Treiber = "PROFI2INTE.vi" (6 von 8)-OK.  
19.02.2004, 12:51:19 : Gerät "PA" - Konfiguration wird geprüft, Treiber = "PA.vi" (5 von 8)-OK.

URL  
C:\dps\Entwicklung\DPS Engine\RESCOURCEN\W0000012\Texte\Film Vortrag.htm

Arbeitsanweisung.....



Prüfung Abbrechen (ESC)    Visualisierung anzeigen    OK (HTML, weiter)    Fehler erfassen

<    >    Login/Logout (F8)    Zurück (F9)    Weiter (F12)    Programm Beenden



# Fehlermanagement

DPS - Fehlermanagement...

**Fehler bearbeiten** Fabrikationsnr  Materialnr  **Miele**

Fehlermeldung

Gerätstatus  Schicht

TxNummer	Materialnr	Fehlerort / Fehlerart
Sonstige Fehler	10 Ohne Teiletausch	IC
	30 Sensoren	

Vorauswahl (abhängig von Fehlermeldung)

Vorauswahl (abhängig von TxNummer)

Vorauswahl für Fehlerort (abhängig von TxNummer und Materialnr)

Funktionsfehler

Vorauswahl für Fehlerart (abhängig vom Fehlerort)

Bemerkung (F8) ...

Teillaufkarte/ Materialanforderung (F9)

Fehler nicht behoben (F10)

Fehler behoben (F11)

Abbrechen (ESC)

- Durch das modulare Konzept mit den dynamischen Einbindungen von Treibern und Prüfabläufen auf Basis von Datenbankinhalten ist eine einfache und nahezu unbegrenzte Erweiterbarkeit bzgl. Treibern, Prüfabläufen und Visualisierungen vorhanden.
- Der Einsatz von LabVIEW ermöglichte die effiziente Umsetzung dieses Konzepts. Das Konzept und die Realisierung wurde von dem NI-Alliance-Partner **A.M.S. Software GmbH** ausgeführt.
- Das Projekt befindet sich derzeit in der Erprobungsphase.
- Mit dem DPS steht Miele ein System zur Verfügung, das die heute bekannten Aufgaben im Prüffeld mit der Integration in die vorhandenen werksinternen EDV-Systeme erfüllt.
- Erweiterungen wie die Integration eines Meßsystems für die Schwingungsanalyse und für die Bildverarbeitung sind bereits in Vorbereitung.



**ENDE**

**Miele**  
**PROFESSIONAL**

■ **Foto**





# Anforderungen und Konzept

- Das Prüffeld wird aus Prüfstationen aufgebaut, die in Ihrer Prüf- und Messtechnikausstattung unterschiedlich sein können aber mit einer einheitlichen Prüfsoftware zu betreiben sind.
- Prüflinge können Baugruppen oder fertige Endprodukte sowie prüfbare Komponenten davon sein.
- Eine Prüfung kann aus mehreren Einzelprüfungen bestehen, die auch auf unterschiedlichen Prüfstationen ausgeführt werden können.
- Es muss in Abhängigkeit von dem Prüfling, dessen Parameter von einer Arbeitskarte eingescannt werden, automatisch über ein Datenbanksystem die aktuelle Prüfung konfiguriert werden.
- Ein intelligentes Fehlermanagement zur Erfassung, Verwaltung und beschleunigter Fehlererkennung und -behebung ist zu implementieren.



# Übersicht DPS – Datenbank-gestütztes Prüfsystem

- Die Daten- und Programm-Verwaltung ist auf einen zentralen Server installiert. Über ein Netzwerk, über das auch die hausinternen EDV-Systeme (Tandem, SAP 91) eingebunden sind, können die einzelnen Prüfstationen angeschlossen werden.
- Prüfgeräte können über die Schnittstelle der SPS oder direkt von einer PC-Schnittstelle angesteuert werden.
- Das Regeln der Betriebsspannung, Umwandeln der Messsignale und die Überwachung der Sicherheitsfunktionen wird von der SPS gesteuert.
- Die Miele-Endprodukte werden über eine serielle Schnittstelle vom PC angesteuert.



# Schnittstellen der EDV - Systeme

- Periodisch wird über SAP 91 ein ASCII-File mit den Materialstamm- und QM-Fehlerklassifizierungsdaten erstellt.
- DPS-Admin läßt die ASCII-File ein und aktualisiert die Datenbestände in der SQL-Datenbank.
- Beim ersten Start einer Prüfung wird durch die eingescannte Fabrikations-Nr. im Prüfsystem aus dem Produktionssystem (Tandem) die Auftragsdaten (Material-Nr., Module, Sprachen, etc.) an das DPS-System gesendet.
- Die Montageabgangsbuchung (MOAB) auf dem Produktionssystem (Tandem) wird bei der letzten Prüfung von dem DPS-System angesteuert.
- Auswertetools können über eine Schnittstelle auf die Prüfprozessdaten (Mess-, Fehlerdaten und Zustandsmeldungen) zugreifen.



- Die Prüfsoftware ist modular aufgebaut und besteht aus den Offline-Komponenten DPS-Admin und DPS-Visualizer und den Online-Komponenten DPS-Engine und DPS-Treiber. Bindeglied und zentrale Komponente ist die DPS-Datenbank.
- DPS-Admin dient zur Vorbereitung von Prüfungen, d.h. Einstellen und Auswahl von Parametern und Prüfabläufen und Ablage in der Datenbank.
- DPS-Visualizer erlaubt das komfortable Ansehen von in der Datenbank gespeicherten Prüfergebnissen.
- DPS-Engine und DPS-Treiber sind für die Ausführung einer Prüfung zuständig.



# Aufbau der Prüfsoftware (Hauptprogramm)

- Die Offline-Prüfsoftware ist mit LabVIEW erstellt und besteht aus zwei Hauptkomponenten, der DPS-Engine und dem DPS-Treiber, die sich wiederum in jeweils mehrere Komponenten aufteilen.
- Die DPS-Engine umfasst alle Funktionen für die Prüfungsdurchführung (Hauptprogramm, Visualisierung, Prozessabbild und Prüfablauf)
- Das Hauptprogramm enthält alle Funktionen, die von der Prüfstation und dem Prüfablauf unabhängig sind. Außerdem werden von ihm die Geräte-Treiber aus der DPS-Treiber-Schicht geladen. Die notwendigen Parameter und Informationen kommen aus der Datenbank.



## **Aufbau der Prüfsoftware (Prozessabbild, Visualisierung, Prüfablauf)**

- **Das Prozessabbild erfasst alle Messwerte einer Prüfung und hält sie in einem Speicher zur Verfügung.**
- **Die Visualisierung macht die Messwerte des Prozessabbildes für den Benutzer sichtbar.**
- **Der Prüfablauf ist für die spezifische Durchführung der aktuellen Prüfung zuständig. Er ist in Form eines Script-Interpreters ausgeführt. Damit muss der eigentlich Prüfablauf nicht fest programmiert werden, sondern kann zur Laufzeit des Hauptprogramms geladen und geändert werden.**



# Aufbau der Prüfsoftware (DPS-Treiber)

- In dem DPS-Treiber sind alle Geräte-Treiber zusammengefasst, die gemäß Ausstattung der Prüfstation dynamisch geladen werden.
- Die Prüfgeräte können entweder direkt von der DPS-Engine oder über die Wandlerschnittstelle in der SPS angesprochen werden.
- Die sogenannten Funktions-Treiber ermöglichen, dass nicht immer die gesamten Treiber für das Gerät im Speicher gehalten werden müssen, sondern nur die benutzten Funktionen daraus.



- Die Parameter einer Gesamtprüfung sind die Sollwerte und eine Parameterliste sowie ein Prüfungsablauf.
- Ein Prüfablauf setzt sich aus Einzelprüfungen zusammen, die ihrerseits aus Prüfschritten bestehen.
- Eine Einzelprüfung findet immer vollständig an einem definierten Prüfplatz statt. Unterschiedliche Einzelprüfungen können auf unterschiedlichen Prüfplätzen durchgeführt werden.
- Die Parametrierung und Zuordnung dieser Abläufe erfolgt mit der DPS-Administratorsoftware, die mit Visual Basic realisiert wurde.



- Die Reihenfolge vom Prüfablauf wird im TreeView-Element entsprechend dargestellt.
- Hierfür wird der AMSL-Interpreter (A.M.S. Scripting Language) benutzt, der Skript-Erstellungen in einer Basic und Pascal ähnlichen Programmiersprache erlaubt.
- Auf der rechten Seite ist ein Ausschnitt von einem AMSL-Prüfskript sichtbar.
- Der Kontextsensitiven Editor für AMSL, ermöglicht in der DPS-Engine im Entwicklungsmodus das interaktive Austesten der AMSL-Skripte.
- Eine Syntaxprüfung und ein einzelschrittfähiger Debugger unterstützen das Austesten.
- Außerdem gibt es eine Hilfe zu sämtlichen in AMSL zur Verfügung stehenden Funktionen, welche von den Treibern generiert werden bzw. dort integriert sind.



# Durchführung einer Prüfung

- Die Durchführung einer Prüfung beginnt mit der Auswahl einer Standardprüfung.
- Über die Fabrikationsnummer können in der Regel alle weiteren Felder der Eingabemaske automatisch mit Daten aus dem Produktionssystem (Tandem) gelesen und eingefügt werden.
- Beim Start der Prüfung werden automatisch die benötigten Treiber und der Prüfablauf (Skripte) geladen und initialisiert.
- Die Ausgabe von Hinweis-Texten ist über HTML-Messageboxen möglich. Dadurch können nahezu beliebige Dateiformate dargestellt werden, so ist bereits daran gedacht, Videos abzuspielen, um den Prüfung deutlicher und einfacher zu beschreiben.
- Weiterhin gibt es einen Entwicklungsmodus und einen Probetriebsmodus. Im Entwicklungsmodus ist es möglich, die zu den Prüfungen gehörigen Skripte zu erstellen und zu bearbeiten.



- Beim Protokollieren der Messwerten wird die Einhaltung von Wertebereichen geprüft und im Fehlerfall automatisch das Fehlermanagement aktiviert. Zusätzlich kann es von Hand aktiviert werden.
- Die Fehler werden gleichzeitig in einer globalen Fehlerdatenbank gespeichert (Messwertfehler bei einer Sicherheitsprüfung).
- Das Abschließen einer Prüfung ist solange nicht möglich, wie Fehler noch nicht behoben wurden.
- Um Fehler im System als behoben zu kennzeichnen, müssen sie automatisch oder manuell klassifiziert werden. Diese Klassifikation ermöglicht eine statistische Auswertung, die z.B. die bisher aufgetretenen Fehler nach Häufigkeit sortiert auflistet.