

# Skripting

## Letzte Woche

1. Umsetzung «praktische Anwendungsarbeit»

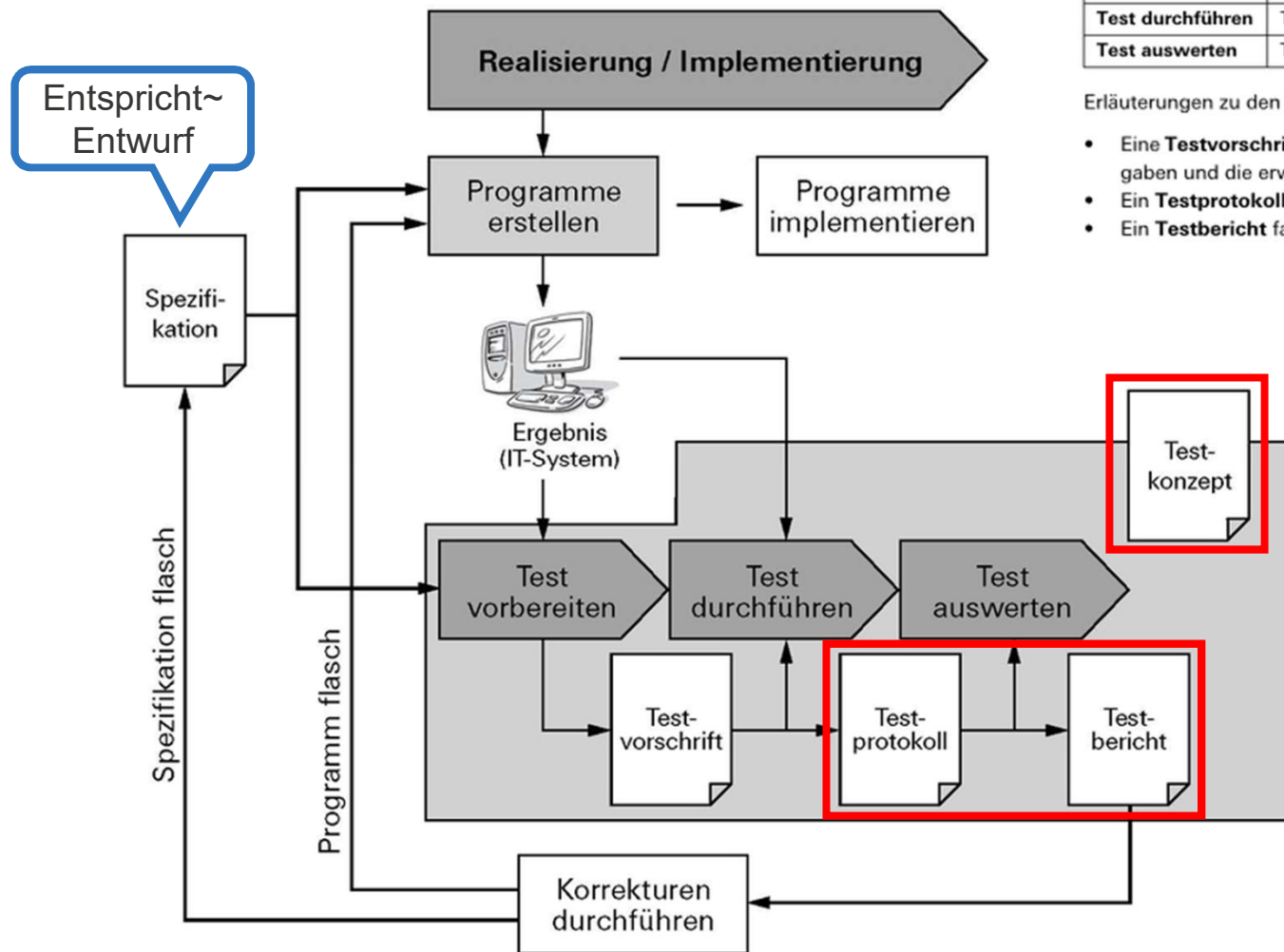
## Diese Woche

1. Fortsetzung Umsetzung «praktische Anwendungsarbeit»
2. Grundlage: Vorgaben aus [Lektion14-Skript S. 1 - 6](#)

Basis-Texte: Alle Texte zu allen Phasen der Skripterstellung (Siehe Sharepoint ab Lektion03)



# Testkonzept, Testprotokoll und Testbericht



Prozessschritte	Inputs	Outputs
Test vorbereiten	Anforderungsspezifikation	Testvorschrift
Test durchführen	Testvorschrift	Testprotokoll
Test auswerten	Testprotokoll	Testbericht

Erläuterungen zu den In- und Outputs:

- Eine **Testvorschrift** ist eine Schritt-für-Schritt-Anleitung, welche die zu tätigenden Eingaben und die erwarteten Ausgaben beschreibt.
- Ein **Testprotokoll** enthält die Ergebnisse nach der Ausführung der Testvorschrift.
- Ein **Testbericht** fasst die Ergebnisse zusammen und zeigt den Handlungsbedarf auf.

# Vorgaben an Entwurf und Test

1. AG liefert Datei: [BackupSpez.txt](#), enthaltend:  
Quell- und Ziel-Verzeichnis des Backups,  
Anzahl zu kopierender Files  
Datei mit Liste aller zu verarbeitenden Files (Backupfiles-List)
2. SL: [Beschreibung der Testfälle](#):
  - a) Was wird getestet
  - b) Testdaten
  - c) Erwartetes Resultat
3. Testprotokoll (Log-File eines Backuptestes) [Testfall01](#) soll enthalten:
  - a) Kompletter Pfad des Quell- und des Zielverzeichnisses
  - b) Anzahl kopierte Files
  - c) Filename Logfile für folgende Datei (Backedupfiles-List)
  - d) Filename aller kopierten Files, in einer separaten Log-Datei
4. Als [Testfall02](#) soll Backupfiles-List und Backedupfiles-List verglichen werden. Wenn beide gleich, war der Backup erfolgreich.
5. Von Testfall01 und Testfall02 ist ein Protokoll (eine Logdatei) zu erstellen. Diese ist im [Testbericht](#) zu erläutern / auszuwerten.
6. Auch AG muss Testfall01 erfolgreich durchgeführt haben

# Vorgaben an «Dokumentation Praktische Arbeit»

1. Muss-Inhalt:
  - a) Teil A: Entwurf
    - 1) PAP-Grafik
    - 2) Wichtige Entwurfsüberlegungen
    - 3) Verwendete **Testpunkte**, um korrekte Funktion nachzuweisen
    - 4) Es ist eine separate Funktion für das Kopieren und eine separate Funktion für das Loggen zu verwenden und zu implementieren!
  - b) Teil B: Uebersicht über Skripte, Funktionen und verwendete **Verzeichnisstruktur** (Quelle und Ziel des Backup)
  - c) Teil C: Test-Protokoll (Log-Datei SL) und Test-Bericht (Auswertung des Test-Protokolls: Nachweis der korrekten Funktionalität d. Backups)
  - d) Teil D: Komplette Skript-Source mit Kommentaren (Teil des pdf-Dokumentes)  
Dokumentation der Verzeichnisstruktur (s. Lektion14)
  - e) Anhang (im pdf): Vorgaben des AG, BackupSpez.txt, inkl. Anzahl zu kopierende Files; Protokoll des durch AG durchgeführten Tests.
2. **Formales** (Abgabe NUR EINES PDF-Dokumentes)
  - 1) Titelblatt (Gruppenname, GL, Mitglieder, Klasse), Seitenzahlen
  - 2) Filename: <GrpNa>\_<GL>\_Dok\_PA.pdf, evt. Mit Version

## Vorgaben an Source

1. Läuft auf VM, unter user admin
2. Verwendet vorgegebene Verzeichnisstruktur, oder eigene Struktur, welche im Dokument dokumentiert ist.
3. Komplette Dokumentation der Source (Kopf, Variabeln, Blöcke)
4. Verwenden von Variabeln
5. Pipelines: Teil-Cmdlets auf Zeilen aufgeteilt
6. Verwenden von Logging (Testprotokolle)
7. Ein- / ausschaltbares Logging
8. Auch «Fehlertests» werden durchgeführt (nicht exist. Verzeichnisse).
9. Source ist konsistent mit Entwurf (bzw. dessen Dokumentation)
10. Guter Programmierstil verwendet (Keine endlosen Anweisungen, sondern schrittweises, zeilenweises Implementieren)
11. Implementation ist flexibel gestaltet und erleichtert Erweiterung (selektiver Backup) in angemessener weise (schrittweises Programmieren: ein Schritt ~ eine Zeile).
12. Implementation ist gut verständlich (Keine Rocket-Science-Statements) und ist wartbar (Dokumentation verständlich für Dritte).

# Bewertungskriterien

## 1. Entwurf

- a) Vollständige Darstellung
- b) Korrekte Verwendung der Symbole
- c) Entwurf und Implementation sind konsistent
- d) Entwurf enthält Testpunkte, welche konsistent mit Testfällen sind  
Testpunkte: Ermöglichen Nachweis der korrekten Funktionalität.

## 2. Implementation

- a) Dokumentation (Kopf, Vars, Blöcke): Vorhanden, einfach verständlich
- b) Verwenden von Variablen
- c) SW (Skript) und Testfälle (Skripts) in 2 Paketen (Skript-Files) erstellt
- d) Verwenden von Logging; Logging ist ein-/ausschaltbar
- e) Guter Programmierstil
  - a) Kurze, einfache Statements
  - b) Code ist einfach verständlich
  - c) Code bietet Hand für Erweiterung

## 3. Test

- a) Korrekte, Verständliche Doku der Testfälle («was», T-Daten, Ergebnis)
- b) Test-Protokoll (Log) erlaubt den Nachweis der korrekten Funktionalität
- c) Test-Bericht (von SL) ist erstellt und ist nachvollziehbar

# Hochladen Dokumente

1. GL lädt Dokument hoch in ClassNoteBook  
3 Verteilte Seiten > SW14\_PA\_uplds