

## 1. Welche Entwicklungsphasen umfassen den Prozess des Protocol Engineering? Welche Ergebnisse liegen am Ende jeder Phase vor?

- Anforderungsanalyse - Was soll das Protokoll leisten?
- Entwurf/Design - Wie soll es das tun?
- Verifikation - Kann es leisten was es soll?
- Implementierung - Fertiges lauffähiges Protokoll (Kode)
- Test - Entspricht Implementierung Design/Entwurf und Anforderungsanalyse
- Integration/Installation - Einrichten und Inbetriebnahme

## 2. Was ist der Zweck der Dienstspezifikation?

- Was der Dienst tun soll
- Art und Weise seiner Nutzung

## 3. Was beinhaltet die Dienstspezifikation?

- Auflistung von (Teil-) Diensten und Dienstprimitiven
- Abhängigkeiten zwischen Dienstprimitiven
- lokales / globales Verhalten
- interne Ereignisse
- Nichtdeterminismen
- Parameter und Abhängigkeiten zwischen Ihnen

## 4. Was ist Nichtdeterminismus? Wie und warum wird er in Dienst- und Protokollspezifikationen genutzt?

- Nicht hundertprozentige Sicherheit dass, das Protokoll in jeder Situation berechenbar reagiert.
- Wird in Kauf genommen, um die **Komplexität** zu reduzieren und die **Flexibilität** zu erhöhen
- Tritt auf bei:
  - Nebenläufigkeit
  - interne Ereignisse
  - Verhaltensalternativen

## 5. Was ist der Unterschied zwischen lokalem und globalem Verhalten in der Dienstspezifikation?

- lokal: Verhalten und Zugriff an einem Service Access Point (SAP) durch Nutzer
- global: Verhalten zwischen verschiedenen SAPs

## 6. Warum wird zusätzlich zur Dienstspezifikation eine Protokollspezifikation benötigt?

- Die Protokollspezifikation beschreibt wie der Service im Detail erbracht wird.
- Sie definiert die Schnittstellen und beschreibt das interne Verhalten.
  - Genauer Ablauf durch Betriebssystem und Ablaufumgebung

## 7. Was muss alles in der Protokollspezifikation beschrieben werden?

- Zustandsautomaten mit Transitionen bei Ereignissen
- Datenformate
- Nichtdeterminismen
- interne Ergebnisse
- lokale Aktionen

## 8. Protokolle können verhaltens- und ablaforientiert beschrieben werden. Erläutern Sie die Vor- und Nachteile beider Beschreibungsweisen!

- **verhaltensorientiert**
  - Wie soll das Protokoll reagieren wenn ein bestimmtes Ereigniss eintritt (Unbestimmte Reihenfolge)
  - Vorteile:
    - \* Verhalten bei verschiedenen Eingaben klar ersichtlich (Unterschiedlich Transitionen)
    - \* Leichter zu Implementieren
    - \* Einfach Darstellung
    - \* Natürliche Beschreibung eines Systems
  - Nachteile:
    - \* ggf sehr groß und unübersichtlich
- **ablaforientiert**
  - Wie läuft eine typische Protokollsitzung ab (Wie folgen die Aktionen aufeinander(Reihenfolge))
  - Vorteile:
    - \* Leichter zu definieren
    - \* Gut für Dienstspezifikation
  - Nachteile:
    - \* Beschränkte Sicht auf das System

**9. Welche formalen Beschreibungsmethoden kennen Sie, mit denen das Protokollverhalten verhaltens- bzw. ablaufforientiert beschrieben werden kann?**

- SDL - beides
- MSC - ablauf
- UML2 - beides (mit MSC und Zustandsautomat)

**10. Erläutern Sie, wie man ein Protokoll mit Hilfe von Endlichen Zustandsautomaten (Finite State Machines) beschreibt! Was sind die Vorteile und Nachteile dieser Beschreibungsmethode?**

- Zustand/State wird definiert - Knoten
- Transitionen werden über Kanten abgebildet
  - Transition auslösendes Ereignis und Ausgabe auf Ereignis wird an Kante beschrieben
- Automat definiert feste Endpunkte oder ist zyklisch ausgeführt.
- Vorteile:
  - Verhalten bei verschiedenen Eingaben klar ersichtlich (Unterschiedlich Transitionen)
  - Leichter zu Implementieren
  - Einfach Darstellung
  - Natürliche Beschreibung eines Systems
- Nachteile:
  - ggf sehr groß und unübersichtlich

**11. Welchen Nutzen hat es, ein Protokoll mit der Specification and Description Language (SDL) zu beschreiben?**

- Vorteile:
  - Fest definierte Beschreibungssprache
  - Objektorientierter Ansatz
  - Sowohl graphisch als auch textuell
  - Es kann Code aus der Beschreibung generiert werden
  - Kann Nebenläufigkeit
  - Beschreibt interne Zustände der *Agenten*
  - Beschreibt Kommunikation der Agenten untereinander

**12. Bei welcher Art von Protokoll ist es Sinnvoll eine Beschreibungstechnik wie die Specification and Description Language (SDL) zu verwenden? Erläutern Sie! (Hinweis: vergleichen Sie folgende Protokolle und treffen Sie dann jeweils eine Entscheidung: UDP, TCP, DHCP, HTTP, RADIUS, Diameter, EAP, TLS)**

- UDP, TCP, EAP, TLS, HTTP - nicht sdl
- DHCP, RADIUS, Diameter
- Bei Aktionen nach außen besser SDL
- Bei Protokollen, die viele interne Zustände und optionale Wege erlauben - SDL

**13. Welche Vorteile bietet ASN.1 bei Design, Implementation und Betrieb von Kommunikationsprotokollen?**

- Nah an der Implementierung
- Gute textuelle Beschreibung im Design - Datenformate
- Datenformate gut beschrieben und gut umzusetzen

**14. Was sind Sicherheits- und Lebendigkeitseigenschaften?**

- Lebendigkeitseigenschaften: Tritt ein Zustand mindestens einmal ein?
- Sicherheitseigenschaften: Ist sichergestellt das fehlerhaftes Ereignis nie eintritt

**15. Was ist das Ziel der Protokollverifikation? Worin besteht der Unterschied zum Protokolltest?**

- Entspricht das Design dem Dienstentwurf und der Anforderungsanalyse
- Unterschied: Testet nicht die Funktionalität einer Implementation
- Verifikation: Rein theoretisch

**16. Warum benötigt man eine Implementierungsspezifikation?**

- Geht auf Besonderheiten der speziellen Umgebung ein
  - Wie werden Prozesse gehandelt?
  - Methoden/Klassen werden verwendet
  - Sprachen

**17. Was sind die Bestandteile der Implementierungsspezifikation?**

- Dokumentation
- Beschreibung der expliziten Ausführung der Implementation auf einem System
- Programm Dokumentation

**18. Was ist das Ziel der Erreichbarkeitsanalyse? Welche Protokolleigenschaften können mit der Erreichbarkeitsanalyse nachgewiesen werden?**

- Prüfen ob alle Zustände im Zustandsautomaten erreicht werden
- Sicherstellen, dass keine Deadlocks auftreten

**19. Was ist ein Test? Welche Aussagen kann ein Test treffen?**

- Test: Spezieller Ablauf einer Protokollinstanz mit vorgegeben Werten - werden erwartete Anforderungen erfüllt
  - Nicht vollständig!
- Gegenstück zur Protokollverifikation
- Soll Fehler finden *NEIN, DOCH, OHH!*

**20. Welche Testurteile gibt es? Wie kommen diese zustande?**

- passed - OK
- failed - kaputt
- inconc - nicht ausreichend spezifizierter Test
- Error - Unerwartetes Ereignis - Softwarefehler - Absturz bei aufruf

**21. Wie wird das Resultat einer Testsuite ermittelt?**

- Durchvorgabewerte für passed - OK
- Aufruf einer Funktion und prüfen des Ergebnisses auf den erwarteten Wert

**22. Es gibt vier Arten des Protokolltests. Nennen und Erläutern Sie diese!**

- Entwicklungsbegleitende Tests

- Debugging
- **Konformitätstest**
  - Übereinstimmung mit Spezifikation
- **Interoperabilitätstest**
  - Zusammenarbeitsfähigkeit von Implementierungen
- **Leistungstest**
  - Leistungsverhalten der Implementierung
- **Robustheitstest**
  - Verhalten der Implementierungen bei falschen Eingaben

### **23. Erläutern Sie die Begriffe Testbarkeit und Design for Testability!**

- Funktionen müssen bereits bei Implementierung auf das Testen ausgelegt sein.
- Es muss ein Testframework existieren, dass die Funktionen entsprechen aufruft und prüft

### **24. Welche Fehlertypen kann ein Endlicher Automat (Finite State Machine) enthalten?**

- Deadlock
  - festhängen in einem Zustand
- Livelock
  - Nicht gewünschte Schleifen
- spontane Transition

### **25. Was sagt die Fehlererkennungsmächtigkeit über einen Test aus?**

- Gibt an wie gut Test sich zur Erkennung von Fehlern eignet
  - vollständige Aufdeckung aller Fehler möglich, oder bleibt eine Unsicherheit nach Test?

### **26. Welche Komponenten enthält die Testarchitektur nach TTCN? Welche Aufgaben haben die einzelnen Komponenten?**

?