

# IT-Modernisierung Braindump

SS 21

5 ECTS

Die Reihenfolge weiß ich zwar nicht mehr, aber es waren 8 Aufgabenbereiche.  
Die Teilaufgaben sind vollständig aufgelistet, Antworten entsprechend markiert.

## 1 IBM Z:

1.1) Digitale Transformation am IBM Z erklären.

Antwort:

- Development Process: Waterfall => Agile => DevOps
- Application Architecture: Monolithic => N-Tier => Micro-Services
- Deployment Packaging: Physical Servers => Virtual Machines => Containers
- Application Infrastructure: Datacenter => Connected => Hybrid Cloud

1.2) 2 Beispiele pro Bereich des IBM Z nennen:

- Sprache, Laufzeitumgebung
- Middleware
- Networking
- Cloud-/Container-Services
- DevOps/Automation
- Big Data, Data Analytics

## 2 Performance-Tuning (itestra):

2.1) Amdahl's Gesetz (Formel für Speed-Up) an einem Beispiel erklären.

2.2) Die 3 Bereiche der Performance-Analyse erklären.

Antwort:

- Tracing: ...
- Accounting: ...
- Profiling: ...

2.3) Performance-Tuning erklären (CPU-Zeit und echte Zeit???)

2.4) Vor- und Nachteile von Sampling und instrumentiertem Code erklären.

- Sampling:

Vorteile:

besser für produktive Umgebungen geeignet

Aufrufe von Systemmodulen können ebenfalls gemessen werden

Nachteile:

Ausführung zwischen zwei Samples wird nicht erfasst, "Samplingrate" ist entscheidend für Genauigkeit

Keine kompletten Callstacks nachvollziehbar, höchstens die aktuelle Aufrufebene ist verfügbar

- instrumentierter Code:

Vorteil: Hohe Genauigkeit (Call-Stacks, Aufrufanzahl etc.)

Nachteile:

Zusätzlicher (die Messung verfälschender) Ressourcenverbrauch, insb. bei kleinen Methoden/Funktionen Ggf. Erfordernis der Neukompilation

Beides ist problematisch/aufwändig in produktiven Umgebungen

### 3 How Assumptions Shape Core Technologies:

3.1) "5 Laws of really good software" nennen.

Antwort:

- Can be written only by very small groups of very skilled programmers.
- Is almost always a labor of love.
- Is almost always begun because the author himself needs it; it makes his computer do something that she really wants it to do.
- Is never finished; it continues to grow. If it doesn't grow, it decays.
- Has a passionate community around it.

3.2) Welche falschen Annahmen über Software werden an Universitäten getroffen?

3.3) Welche Annahmen sollten für Software getroffen werden?

Antwort:

- Manageability beats features.
- Ease of use beats speed and features.
- Well documented interfaces are key to long term survival of product (nothing beats IBM documentation).
- Your constraints will be defining characteristics of your software.
- Build and it, and they will come – far out features will become common-practice in the near future (e.g. Live Migration).
- Thinking about what assumptions are being made and validating them before designing and writing first line of code is key.

## 4 Modernisierung (itestra, T-Systems):

4.1) Die Begriffe Alterung von Außen, Heterogenität und Wissensverlust erklären.

Antwort:

- Alterung von außen: Fachprozesse ändern sich, neue Technologien eröffnen Möglichkeiten und wecken Erwartungen (Mobil, Online, 24/7, sofortige Reaktion...).

Erhöhte Anforderung durch Erfolg: Erhöhte Volumina (z.B. neue Kunden), Höhere Reportingfrequenz (stündlich statt monatlich).

- Heterogenität: vmtl. bzgl. Sprachen/Stilen.

In der Praxis unvermeidbar:

1. Lange Lebensdauer (alles neu alle 10 Jahre ist unmöglich)

2. Unternehmensfusionen etc.

3. Persönliche Präferenzen, Hypes, fehlende Strategie

Probleme: Technischer Support / Lauffähigkeit, Verfügbarkeit Attraktivität für Mitarbeiter Veraltete Werkzeuge, Libraries => Produktivität

- Wissensverlust: Fluktuation des Entwicklerteams, i.d.R. keine adäquate Dokumentation.

"keine Zeit", Im Rahmen von Projekten nur Delta-Dokumentation, Typisch: Word-Dokumente auf Netzlaufwerk

4.2) Erklären was an Wrapping, Neuentwicklung und Kaufen (Buy) schlecht ist.

Antwort:

- Wrapping: Unmodifizierte Weiterverwendung über Fassaden. Verfestigt die Innovationshindernisse! Schwache Performance.

- Buy: Hohe Kosten/Zeitbedarf, i.d.R. viel Customizing = SW Entwicklung. Keine Differenzierung! Garantiert die Lösung des Anbieters höhere Agilität für die Zukunft?

- Neuentwicklung: Fortschritt möglich, aber typische Risiken eines großen Softwareprojekts (Requirements Creep, Zeit/Kosten, Big Bang). Ist-Fachlogik i.d.R. undokumentiert => Neukonzeption.

4.3) 3 Big Bang Ansätze erklären.

Auf der entsprechenden Folie konnte ich nur 2 finden: Green Field Neuentwicklung, sowie Re-Hosting und Konvertierung

4.4) 3 Vorteile von Virtualisierung nennen.

4.5) Flexibilisierungsstrategie im Bezug auf Legacy-System erklären.

## 5 Gemischtes weil keine Ahnung mehr:

5.1) 4 Skill-Profile mit einer Skizze erklären.

Die Lösung hierzu befand sich auf einer von 132 Folien aus dem entsprechenden Vortrag, ganz am Ende. Und es war eine "Leseempfehlung" eines weiterführenden Artikels.

Antwort:

- "i"-shaped: historical
- "T"-shaped: bare minimum
- "pi"-shaped: adavantageous
- "comb"-shaped: goal

5.2) Den Begriff Data Gravity erklären. (Es war eine Folie, mit 5 Buzzwords)

Antwort:

- Daten ziehen Anwendungen an
- Zentrale Datenhaltung
- Stammdaten
- Quelle der "Wahrheit"
- Kerndatenverarbeitung
- Transaktionale Workload.

5.3) Je ein Cloud Integration Pattern der 4 Bereiche Process Centric, Application Centric, Data Centric Integration und Event Driven erklären. (riesige Walls of Text)

5.4) Begriff ACID von Datenbanken erklären.

Antwort:

- Atomicity: eine Transaktion wird ganz oder gar nicht durchgeführt.
- Consistency: eine DB ist vor und nach einer Transaktion in konsistenten Zustand.
- Isolation: parallele durchgeführte Transaktionen beeinträchtigen sich nicht gegenseitig.
- Durability: Das Ergebnis einer Transaktion in der DB muss persistent sein.

5.5) Vor- und Nachteile von statischem und dynamischem SQL erklären.

5.6) 6 Ziele nennen von Unternehmen, die auf Cloud setzen.

5.7) Zeichnung über Cloud aus Folie gegeben mit Lücken. Off-Premise und On-Premise stehen dort.

Antwort:

- Private bei der On-Premise Seite.
- Public und Dedicated bei Off-Premise.
- Distrbuted dazwischen auf der Grenzlinie.

5.8) Welche Vorteile erhofft man sich von Z DevOps auf dem Mainframe?