Nicht Funktionale Anforderungen bzw Safety Req .

Daniel Lammering An: Julian02 Brand, Maximilian Brueckl, Sebastian Brunthaler 11.07.2018 14:26

Protokoll: Diese Nachricht wurde beantwortet.

Hi,

Ihr könnt es ja auch als Tabelle aufbauen und dazu dann Requirements als Ausgangsbasis her nehmen und diese dann mit Mechanismen beschreiben.
Ich stell mir das so bissl vor:

Art des Anforderung | Beschreibung | Parameter/Absolute Zahlen | Metrik /Maßstab | Auswirkungen auf Kommunikation | Technische Lösung um Req. zu erfüllen.

Bzgl non-functional requirements:

A functional requirement describes *what* a software system should do, while non-functional requirements place constraints on *how* the system will do so.
The functional requirement is **describing the behavior of the system** as it relates to the system's functionality. The non-functional requirement **elaborates a performance characteristic** of the system.

Typically non-functional requirements fall into areas such as:

- Efficiency
- Effectiveness
- Extensibility
- Fault tolerance
- Reliability
- Scalability
- Response time
- Perfomance
- Robustness
- Real Time with short latency period
- fault detection
- fault- tolerant
- redundant-determinism

Wie ihr die dann "Messbar" macht kommt immer drauf an.

Performance könnte man messen anhand der Response Time. oder der Latenz
Efficiency: minimal use of resources (memory, processor, disk, network…)
Robustness: in the presence of faults, stress, invalid inputs
Scalability: for large number of users or quantities of data

Reliability wird meistens mit der Mean Timt to Failure (MTTF) beschrieben:
Measure degree to which the system performs as required
•Includes resistance to failure
• Ability to perform a required function under stated conditions for a specified period of time
• Can be measured using
• Probability that system will perform its required function for a specified interval under stated conditions

• Mean-time to failure

Availability kann auch in % angegeben werden. Wenn ihr jetz Rekonfiguriert, ist es ja nicht verfügbar.
Beispiel:
Definition: Percentage of time that the system is up and running correctly •

Can be calculated based on Mean-Time to Failure (MTBF) and Mean-Time to Repair (MTTR) •
MTBF : Length of time between failures
MTTR : Length of time needed to resume operation after a failure
Availability = MTBF/(MTBF+MTTR)

=> May lead to architectural requirements:
• Redundant components (lower MTBF)
• Modifiability of components (lower MTTR)
• Special types of components (e.g., self-diagnostic)
Measurement: The mean time to failure and mean time to repair of critical components must be identified (typically measured) or estimated

- Examples
  - The system shall meet or exceed 99.99% uptime.
  - The system shall not be unavailable more than 1 hour per 1000 hours of operation.
  - Less than 20 seconds shall be needed to restart the system after a failure 95% of the time. (This is a MTTR requirement)

Availability	Downtime
90%	36.5 days/year
99%	3.65 days/year
99.9%	8.76 hours/year
99.99%	52 minutes/year
99.999%	5 minutes/year
99.9999%	31 seconds/year

Hier noch mehr Details zur Fehlerbehandlung:

The development of a dependable computing system calls for the combined utilization of a set of four techniques:

1. fault prevention: how to prevent the occurrence or introduction of faults,
2. fault tolerance: how to deliver correct service in the presence of faults,
3. fault removal: how to reduce the number or severity of faults,
4. fault forecasting: how to estimate the present number, the future incidence, and the likely consequences of faults.

Fault handling prevents located faults from being activated again. Fault handling involves four steps:

• fault diagnosis, which identifies and records the cause(s) of error(s), in terms of both location and type,
• fault isolation, which performs physical or logical exclusion of the faulty components from further participation in service delivery, i.e., it makes the fault dormant,
• system reconfiguration, which either switches in spare components or reassigns tasks among nonfailed components,

• system reinitialization, which checks, updates and records the new configuration and updates system tables and records

Mit freundlichen Grüßen/Best Regards,

Daniel Lammering
Vehicle System Architect
Vehicle Systems Architecture - Future Architecture Solutions

Cross Divisional Systems & Technology
Corporate Systems & Technology

Continental
Corporate Functions

Besucheranschrift / Visitor address :
Continental Automotive GmbH
Osterhofener Str. 10, D-93055 Regensburg

Rechnungsanschrift / Invoice address :
Continental Automotive GmbH
Siemensstrasse 12, D-93055 Regensburg

Tel.: +49 941 790 5914
Mobile: +49 151 2030 6551
E-Mail: daniel.lammering@continental-corporation.com

<$$014!>