Ickefunktionella krav

Vid upphandling av verksamhetsstödjande system

Innehållsförteckning

0.1 Syfte med dokumentet 3

0.2 Versionshistorik 3

0.3 Referenser 3

1 Inledning 4

## Syfte med dokumentet

Dokumentet beskriver gemensamma, icke-funktionella krav som referensarkitekturen ställer på upphandlade lösningar. Skall användas vid upphandlingar.

## Versionshistorik

| Datum | Version | Utfärdare | Förändringsorsak |
| --- | --- | --- | --- |
|  | 0.1 | Johan Eltes | Utkast med tänkbara rubriker. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Referenser

[0]

# Inledning

## Syfte

## Bakgrund

# Generella krav

# Användbarhet

[This section should include all of those requirements that affect usability. Examples follow]

## Utbildningsinsats

[Describe the time for a normal users and power users to become productive at particular operations]

## Measurable task times for typical tasks

[The requirement description.]

## Requirements to conform to common usability standards

[The requirement description, for example, IBM’s CUA standards or Microsoft’s GUI standards]

{Include accessibility requirements, keyboard use, mouse use, other input devices}

## <Another Usability Requirement >

[The requirement description.]

# Reliability

[Requirements for reliability of the system should be specified here. Suggestions are as follows:

## Availability

[Specify percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and the like.]

{When can the system be unavailable (preferred maintenance window times/days, etc)? When must it be available? Are there any functions of the system which can be less available then others (functions which can be unavailable for a time period without crippling production use)? What are the functions which are most critical to be available?}

## Mean Time Between Failures (MTBF)

[This is usually specified in hours but it could also be specified in terms of days, months or years.]

## Mean Time To Repair (MTTR)

[How long is the system allowed to be out of operation after it has failed?]

{Max block of time a function can be out of operation. Identify per high level functions, or if there is a specific time constraint on a certain function}

## Accuracy

[Specify precision (resolution) and accuracy (by some known standard) that is required in the systems output.]

{At a global level of the system - Date vs. Date&Time (Time to sec or msec) for calendars, number of significant digits for certain units (no fractions for some units, tenths, hundredths etc for others). Currency always to Kr/Euro/USD. Accuracies unique to a specific functionality should be addressed in associated use-case.}

## Defect Rate

[The requirement description could be expressed in terms of maximum number of bugs per bug classification in total and per functional area but could also be expressed as bugs/KLOC (thousands of lines of code) or bugs/function point.]

{For risk analysis, what aspects of the system are business critical and can not have any logic or data defects (such as order definition, etc) and what areas may have minor defects (such as usability)?}

## Bug Classification

[Bugs are categorized in terms of minor, significant, and critical bugs. The requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or complete inability to use certain parts of the functionality of the system.]

## <Another Reliability Requirement >

[The requirement description.]

# Performance

[The performance characteristics of the system should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name. The following are performance categories to consider:]

## Response time for a transaction (average, maximum)

[The requirement description.]

{Itemize for major functional areas, providing expected response time for an operation from the system use point of view (Use Case step). Identify any unique time constraint for any functional area (Ex: process a batch operation X must complete in one hour because there can be 24 such batch operations which must complete in one day).}

## Throughput

[The requirement description, for example, transactions per second/day/week/month]

{Nominal and peak throughput for major functional areas or functional areas where there may be a unique throughput constraint. If system does not operate 24 hrs/day then calc in smaller unit (hours,etc).}

## Capacity

[The requirement description, for example, the number of customers or transactions the system can accommodate]

{Identify unique capacity constraints due to some business condition like end of month, etc. Identify number of users per functional area of appropriate. Some details such as max batch size or number of options in a car order are left to use case}

## Degradation modes

[What is the acceptable mode of operation when the system has been degraded in some manner]

{Section 5.1 should completely cover for all major areas.}

## Resource use

[Document memory, disk, communications, etc.]

## Estimated number of tables

[The requirement description. UJ: This should not be regarded as a requirement. This is background information that is of interest for the architect since it affects decisions regarding capacity planning and moving data from the old system(s) into the new one.]

{Estimate from existing system but may not be applicable}

## Estimated production database disk storage

[Preferably based on volumetric. UJ: See previous note under 6.6]

{Estimate from existing system but may not be applicable. Identify any known unique constraint which may obviously effect}

## (One-time) Data conversion requirements

[The requirement description.]

{Known issues with 'dirty' data should be identified so one time cleanup can be addressed (Ex: legacy system bug resulted in two identifiers for same thing, would like to make one.. Purge out test data, etc). Desire to change or standardize format of data (Would like common date format for old data). }

## Frequency/volume of ongoing data loads/refreshes

[The requirement description.]

{Any current periodic ETL (Extract Translate Load) type operations identified}

## Capacity Projections

[Planned increases in number of users, space requirements]

{Projections for capacity on major operations (number of users, order throughput, size of batch jobs, etc) for next 5, 10 etc yrs. If there is data from existing system, provide that as well. Consider capacity requirements if system were to be used by largest plant (north america?) in company, etc}

## <Another Performance Requirement>

[The requirement description.]

# Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.]

## Data backup and archival requirements

[The requirement description.]

{Check if existing system requirements are valid for proposed system }

## Disaster recovery classification

[The requirement description.]

{ Check if existing system requirements are valid for proposed system }

## Existing hardware/software planned to be used with this system

[The requirement description for clients/servers.]   
[NG: For exemple requirements on the target environment.  
Exemple

* The client portion of the system shall operate on any personal computer with a 486 processor or greater. The client portion shall require less than 20 MB disk space and 32 MB RAM.]

{Any reallocation from existing system, such as network, clients, etc? }

## <Another Supportability Requirement >

[The requirement description.]

# System Constraints

[This section needs to indicate any system constraints on the system being built.]

## Design Constraints

[This section needs to document design decisions that have mandated and must be adhered to. Examples include software languages, software process requirements, architectural and design constraints, purchases components, class libraries, frameworks and so on.]

{ Check if existing system requirements are valid for proposed system }

### <Design Constraint One>

[The requirement description.]

## Interface Constraints

[This section needs to document any constraints on Interfaces that the system must have to talk to external entities (Legacy Systems, API's) etc.]

{Identify interfaces to external systems related to proposed functionality. An example would be an Authentication system which is accessed by the planned decommissioned system and required to be used by the proposed system.

For each interface, provide as much of below as possible:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes (Other Value Described)** |
| Interaction Scheme | Remote Procedure call/Reqest-Response/Datagram/Publish-Subscribe/Multicast/FileTransport/Share |  |
| Transport | HTTP/MessageQ/FTP/WebService/Email |  |
| Security | Authenticated/Encrypted |  |
| Availability | Not Guaranteed/Guaranteed Delivered/Guaranteed In Sequence |  |
| Latency | Determistic/variable |  |
| Throughput | Unit of work per sec, etc |  |
| Lifecycle | Stateful conversation/Stateless |  |

}

### <Interface Constraint One>

[The requirement description.]

## Environment Constraints

[This section needs to document any constraints on development tools, hardware and software platforms that are required to support the system.]

{Restrict to client system- browser requirements, printer requirements, client memory. Somewhere in this should be compatibility like screen resolution but not here}

### <Environment Constraint One>

[The requirement description.]

## Decommissioning Constraints

### External Interfaces used to support proposed functionality

{Refer to section 8.2. If this is not a subset of section 8.2, clarify}

### Services supplied but outside of proposed functionality

{ For all decommissioned systems, identify interfaces to services supplied but NOT related to proposed functionality. This information is captured to manage scope of the project.}

### Shared resources used to support proposed functionality

{For all decommissioned systems, identify resources shared between them and external systems used to support proposed functionality. Examples may be a Network Attached Storage device, Email server, printer. }

### Resources used but outside of proposed functionality

{ For all decommissioned systems, identify resources shared between them and external systems UNRELATED to proposed functionality. This information is captured to manage scope of the project.}

# Online User Documentation and Help System Requirements

[Describes the requirements, if any, for on-line user documentation, help systems, help about notices, and so on.]

{Anything unique such as mouseless, accessibility requirements, etc }

# Purchased Components

[This section describes any purchased components to be used with the system , any applicable licensing or usage restrictions, and any associated compatibility/interoperability or interface standards.]

# Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

# Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notice, wordmark, trademark, or logo compliance issues for the software.]

{global- appears on UI, printed paper, email sent, etc}

# Applicable Standards

[This section describes by reference any applicable standards and the specific sections of any such standards that apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

# Application Profile

[This section describes the profiles of various users and contacts for external systems.]

## User Profiles

|  |  |  |  |
| --- | --- | --- | --- |
| **User Type** | **Locale** | **Max. Users** | **Max. Concurrent Users** |
| *[Run/View Reports]* | [Metro Detroit] | [50] | [10] |
| *[Data Entry]* | [Europe] | [20] | [10] |

## Sources of data and contacts (External systems)

|  |  |  |
| --- | --- | --- |
| **Source** | **Platform** | **Contact** |
|  |  |  |

## Targets of data and contacts (External systems)

|  |  |  |
| --- | --- | --- |
| **Target** | **Platform** | **Contact** |
|  |  |  |

# Krav vid systemutvecklingsentreprenad