cetorp.dk

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

# Calculation of Risk Factor

## Using the Excel spreadsheet

## Calculation of Risk Factor.xls

⇩ Table of Contents

## Events, Impact and Software Validation

Many software products in complex computer systems like LIS or LIMS involve a potential risk that some adverse events may have an impact on the companies using the software. Events may be caused by software errors or by actions causing the software to behave unexpectedly and in a faulty manner. A way to prevent impact is to identify the most risky error events in the computer system and then perform a risk-based validation of the associated software in order to avoid or reduce the effect of fault conditions. The *Risk Factor calculation spreadsheet* is a tool that may help to estimate the severity of such risks and to prioritize of the software validation efforts with respect to scope and depth.

The calculated risk factors can only be used in this context and are not meant as a general measure for risks caused by software malfunction and errors. It is simply meant to point out those software products in the computer system that most certainly require proper software validation in order to prevent, or at least reduce, the impact caused by identified error events.

## The Risk Calculation Sheet

The Excel workbook may contain as many sheets with identical calculation forms, as needed. Each form represents a software product for which a risk factor can be estimated. The upper part of the form is reserved for description and comments, while the lower part is a table used to tick off weighted risk probability scores for automatic calculation of the risk factor.

| Risk Calculation Form | | | | | | | | Risk factor = 0 | Consequence |
|---|---|---|---|---|---|---|---|---|---|
| Software product: | | | | | | | | | |
| Software used for: | | | | | | | | | |
| Comments: | | | | | | | | | |
| A: | | | | | | | | | |
| B: | | | | | | | | | |
| C: | | | | | | | | | |
| D: | | | | | | | | | |
| E: | | | | | | | | | |
| F: | | | | | | | | | |
| Made by: | Date: | | | Approved by: | | | Date: | | |
| **A - Software category** | Category 1 | | (1 p) | Category 2 | | (2 p) | Category 3 | (4 p) | 0 |
| **B - Interaction with input** | Type 1 | | (1 p) | Type 2 | | (2 p) | Type 3 | (3 p) | 0 |
| **C - Interaction with output** | Type 1 | | (1 p) | Type 2 | | (2 p) | Type 3 | (3 p) | 0 |
| **D - Internal impact** | High probability | | | Medium probability | | | Low probability | | 0 |
| High impact | | | (9 p) | | | (6 p) | | (4 p) | |
| Medium impact | | | (6 p) | | | (4 p) | | (2 p) | |
| Low impact | | | (3 p) | | | (2 p) | | (1 p) | |
| **E - External impact** | High probability | | | Medium probability | | | Low probability | | 0 |
| High impact | | | (9 p) | | | (6 p) | | (4 p) | |
| Medium impact | | | (6 p) | | | (4 p) | | (2 p) | |
| Low impact | | | (3 p) | | | (2 p) | | (1 p) | |
| **F - Probability of detection** | High probability | | | Medium probability | | | Low probability | | 0 |
| Systematic error | | | (1 p) | | | (4 p) | | (5 p) | |
| Periodic error | | | (1 p) | | | (3 p) | | (4 p) | |
| Sporadic error | | | (1 p) | | | (2 p) | | (3 p) | |

| Calculated risk factor = A + B + C + (D + E) * F = 0 | |
|---|---|

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

The calculated risk factor (which is a number between 0 and 100) is a relative quantity that only has meaning when compared to other factors estimated on the same basis. The actual risks should always be estimated using the *system perspective* (i.e. estimated relative to the actual system, not a particular process in the system) since that will provide the best basis for comparison.

All sheets use the same score, and each score value is obtained by reference to a similar value stored in a hidden sheet named "Basis". Thus, if a basic score value is changed the calculation on all sheets will change accordingly. All sheets are write-protected (without password) so that only the description area and the yellow tick off cells can be altered. The "Template" sheet is intended as a template for the "Copy and move..." function which must be used to create additional identical Risk Calculation sheets.

## Description and comments

| | | | |
|---|---|---|---|
| **Software product:** | Unique name of the software product or module being risk assessed. | | |
| **Software used for:** | Brief description of what the software product or module is used for. | | |
| **Comments:** | General notes - e.g. Estimated error event: Error occurs when ... | | |
| **A:** | **Software category** | - e.g. Customized standard software ... | |
| **B:** | **Interaction with input** | - e.g. Input from operator and another module ... | |
| **C:** | **Interaction with output** | - e.g. Output to operator and database ... | |
| **D:** | **Internal impact** | - e.g. Low probability and medium impact because ... | |
| **E:** | **External impact** | - e.g. High probability and medium impact because ... | |
| **F:** | **Probability of detection** | - e.g. The systematic error is detected immediately due to ... | |
| **Made by:** | **Date:** | **Approved by:** | **Date:** |

## A: Software Category

The software category is divided into three levels as defined in the table below. Most software modules in a LIMS system belong to Category 2 while software products, which are customized or developed by the users themselves (such as spreadsheets) belong to Category 3. Operative systems and MS® Office packages are normally Category 1.

| Category | Description |
|---|---|
| Category 1 * Standard Software Packages | Commercial off-the-shelf (OTS) software packages. *Examples:* Excel spreadsheets and PC-controlled instruments with minimum configuration. |
| Category 2 Custom Configurable Software Packages | Typical features of these systems are that they permit users to develop their own applications by configuring predefined software modules and by developing new application software modules. *Examples:* Human Machine Interfaces (HMI), Supervisory Control and Data Acquisition (SCADA), Laboratory Automation Systems (LAS), Material Requirements Planning Systems (MRP) and **Laboratory Information Manufacturing Systems (LIMS).** |
| Category 3 * Custom Built Software | Includes any application, off-the-shelf software or other software products that are modified or developed according to custom requirements. This also applies to Standard Software Packages used to develop custom applications and to programming languages. |
| ** Complex spreadsheets with macros belong to Category 3.* | |

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

## B & C: Interaction with Input & Output

Interaction in this context indicates that there is a risk whenever data are transferred to (output) or from (input) other software. E.g. data may be received incorrectly or may be corrupted, or the receiver may respond inappropriately or unexpectedly. It should also be taken into account if data can be re-transmitted in case of errors and that the operator may be part of the risk, even if the data transfer is entirely controlled by software. It is important to make a distinction between input and output.

To make the estimate easier, the risks are divided into three *types* based on the amount and importance of the transferred data.

| **Interaction with Input & output** | | |
|---|---|---|
| **Type 1** | Very few data and/or insignificant contents | Low risk |
| **Type 2** | Certain amount of data and/or rather important contents | Medium risk |
| **Type 3** | Large amount of data and/or very critical contents | High risk |

## D & E: Internal & External Impact

Impact is a measure of how severe and harmful a possible software error event is for the company. This indefinable quantity cannot stand alone, but has to be combined with the probability of the event to occur. An event may be an unexpected behaviour caused by a software error, but may also be an operator action that makes the software behave in an unexpected or faulty manner.

The probability of a severe error event is normally specified by its frequency which may be defined as once per number of transactions or as once per time interval.

| **Probability (frequency) of error event** | | |
|---|---|---|
| **High** | Occurs quite often | E.g. once per 100 transactions or once per day |
| **Medium** | Occurs frequently | E.g. once per 1.000 transactions or once per month |
| **Low** | Occurs seldom | E.g. once per 10.000 transactions or once per year |

An impact may have both internal and external effect...

- **Internal** - e.g. impact on production, profitability, employee satisfaction etc.
- **External** - e.g. impact on regulations, reputation, customer satisfaction etc.

And the risk may thus be estimated as an...

**Internal impact** with

- High effect - e.g. severe impact if the production is stopped in several days
- Medium effect - e.g. modest impact if just a single report has to be recalled
- Low effect - e.g. minimal impact if only a few analyses have to be repeated

**External impact** with

- High effect - e.g. severe impact if requirements from authorities cannot be met
- Medium effect - e.g. modest impact if the authority's action is limited to issue a warning
- Low effect - e.g. minimal impact if only a single customer becomes discontented

In addition, it should be considered if there is a time-dependent aspect associated with the impact, e.g. if there is a significant long-term effect to take into account after if the damage has been repaired.

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

As a main rule, the probability and the effect of impact should always be estimated in the *system perspective* (i.e. estimated relative to the entire system), even if the software is part of a larger process or data-flow.

The probability of impact is generally the same for the *internal* and *external* estimate, but in some cases there may be a difference because the effect is estimated in the system perspective. A certain error could be regarded externally as damaging to the business, but internally as a harmless incident.

## F: Probability of Detection

The consequence of an event may depend on the probability of detecting the occurrence. E.g. a minor failure that is difficult to detect may cause far more damage than a severe error which is detected immediately.

| **Probability of detection** | | |
|---|---|---|
| **High** | Highly likely | E.g. each time the event occurs |
| **Medium** | Reasonably likely | E.g. each time or every two times |
| **Low** | Almost unlikely | E.g. less than every two times |

It is common practice to assume that the probabilities of systematic software errors are high but it is not evident that systematic errors should be detected immediately. Less regular occurrences of software errors are denoted *periodic* and *sporadic*. Such errors can be more difficult to detect, but may cause less damage since they occur less frequently.

In order to take this aspect into consideration, the risk score for the probability of detection is weighted by the frequency of occurrence - a high score if systematic, a medium score if periodic, and a low score if sporadic occurrence.

| **Probability of detection weighted by Frequency of occurrence** | | | |
|---|---|---|---|
| **Systematic** | Each time | (High) | E. g. a program error or data transfer failure |
| **Periodic** | Periodically | (Medium) | E. g. may depend on certain users or time of operation |
| **Sporadic** | Once in a while | (Low) | E. g. random error - could be a hardware problem |

The weighting may not be linear and the cells in the form are therefore assigned individual values.

## Calculation of the Risk Factor

The formula for calculation of the risk factor is

**0 ≤ Calculated risk factor = A + B + C + (D + E) * F ≤ 100**

If the original basic score values are used in the calculation, the result will be a number between 0 and 100. The medium value (all estimates set to medium) is 30 and the lowest reasonable value is 5.

The values in the right-most columns are the risk contributions for each of the impact categories A - F and are termed *consequence* since they indicate the significance of the individual risk categories. If the consequences for different events are compared with each other, a risk pattern could be revealed and hereby give hints to risk mitigation strategies.

E.g. if a computer system is estimated to have many relatively small risks with dominant external impact consequences, it might be at good idea to focus on the cause of this condition and find out if special software validation is required.

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

**Risk factor in short**

- Formal: $0 \leq$ Risk factor $\leq 100$ (when using original basic score values)
- Risk factor *highest* value = 100
- Risk factor *medium* value = 30
- Risk factor *lowest* value = 5
- Risk factors are *relative* to the *system perspective*
- Risk factors must be used with care
- Risk factors are used for ...
  - Feasibility studies
  - Requirements specification
  - Software validation
  - Software updating
  - Performance optimizing

⇩ Table of Contents

## Software Validation

Most computer systems in use are basically tested and verified to a certain extent. The testing is an important part of, but not the conclusion of, the software validation.

Software validation confirms by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled.

While testing basically is performed on software modules to find and repair functional errors, the validation is performed in system perspective, meaning that the validation also takes into account surroundings and users. Validation is concerned with the interaction between the system (hardware, software, personnel, and surroundings) and its stakeholders (owners, customers, and authorities), and is establishing that the system functions satisfactorily and suitably.

Many factors that pose a threat to the computer system are not included in the testing simply because they were never thought of as potential risks. Unfortunately, some of these issues may also be missing in the software requirements specification. The risk calculation task described in this document may thus be used preventively or retrospectively to complete the requirements specification.

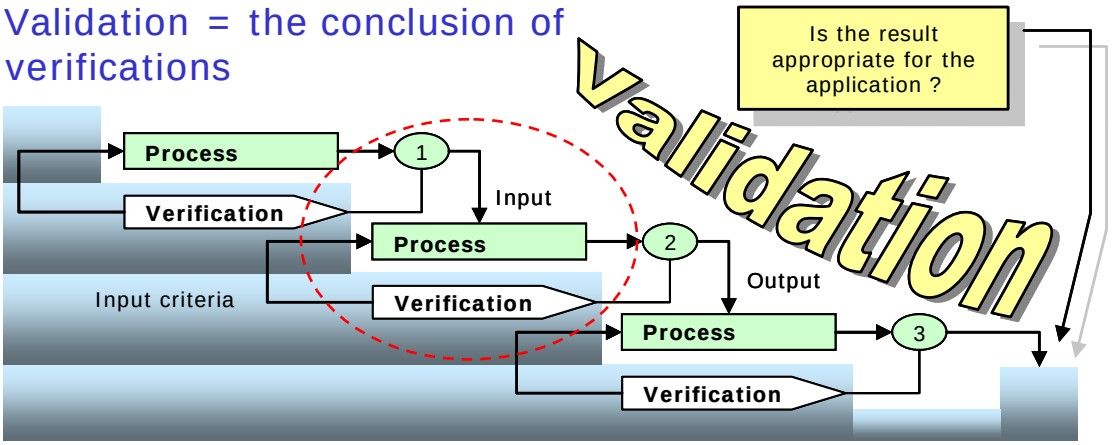
A documented software requirements specification provides a baseline for both validation and verification. The software validation process cannot be completed without an established software requirements specification.

## Process Verification and Validation

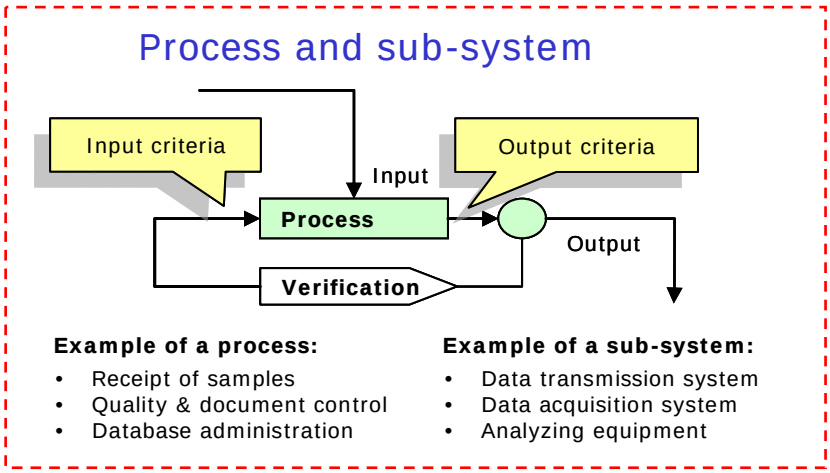
Inside the computer system any activity, which is controlled or assisted by software in order to enable the transformation of inputs to outputs, may be considered as a process. The advantage of this process approach is the understanding it provides for the linkage between the individual processes within the computer system, as well as for their combination, interaction and mutual risks.

The term verification is often used to denote the test and examination carried out to check that an individual process functions as stated in the requirements specification and in the test requirements. Verification is referred to as validation when it is carried out for individual processes in *system perspective* or when it is the *conclusion of the verifications* of the totality of processes combined to constitute the complete computer system.

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark




A sub-system may be a well-defined or self-contained part of the computer system and may commonly be related to separate software modules and hardware units. The sub-system itself may consist of a number of processes, each of which controlling their own transformation of inputs to outputs.




There are many events that can cause a process to fail or produce incorrect results. The verification will commonly prove that if the process receives valid input data, it will produce correct and valid output data. However, the combination of validity, correctness and processing of data may be quite difficult to evaluate as seen in the table below.

| Input from previous | Process | Output to next | Comments | Impact |
|---|---|---|---|---|
| Correct and valid | Ok | Valid | Process verified and approved | None |
| Correct and valid | Failure | Alarm | Correct and valid input rejected | Medium |
| Correct but invalid | Ok | Alarm | Invalid but correct input rejected | Low |
| Correct but invalid | Failure | Valid | Invalid but correct input processed | Medium |
| Incorrect and invalid | Ok | Alarm | Process verified and approved | None |
| Incorrect and invalid | Failure | Valid | Bad input processed | High |
| Incorrect but valid | Ok | Valid | Valid but incorrect input processed | Medium |
| Incorrect but valid | Failure | Alarm | Valid but incorrect input rejected | None |

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

The validity of data is determined by the specified input and output criteria while the correctness and processing of data is related to the actual scenario.

## Risk Scenarios

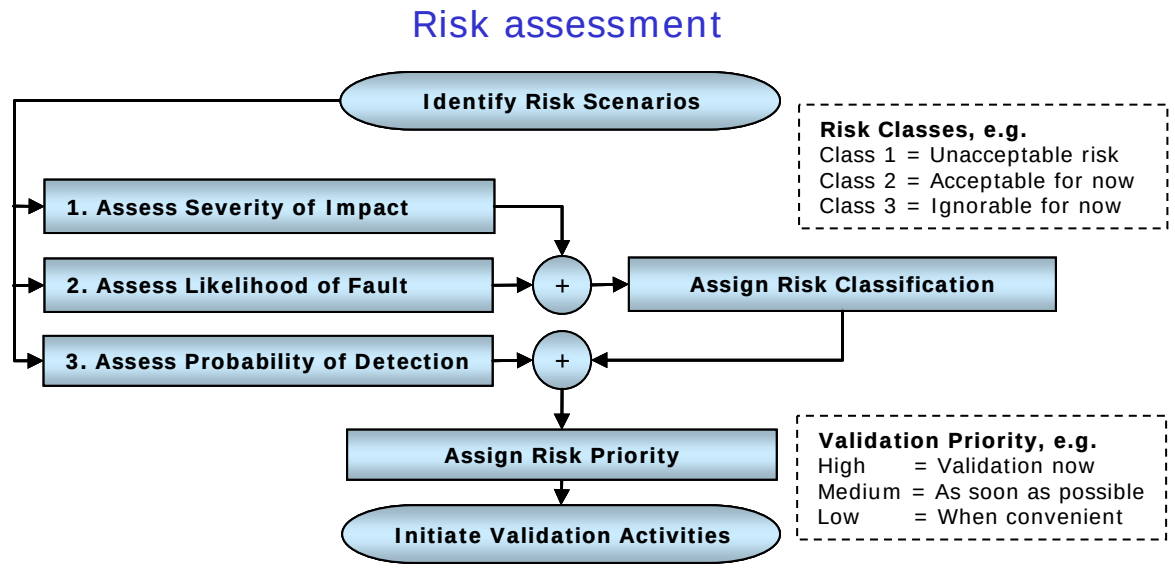
Any process that fails will pose a risk to the company, so the main task will be to point out those particular processes (functions or sub-functions) that have potential risks associated with them, and then proceed to identify the various risk scenarios associated with events that may cause impact due to process failure.

| Process | Function | Risk Scenarios (Events) | Effects | Impact |
|---|---|---|---|---|
| Reception of samples for laboratory analysis | Booking in samples using a bar-code reader on sample labels | Sample sent to wrong analysis instrument | Sample not accepted by the system, sample will be redirected and the analysis only slightly delayed | None |
| | | Bar-code read incorrectly | Sample not accepted by the system and analysis postponed | Low |
| | | Customer address out of date | Sample accepted by the system and report and invoice postponed | Medium |
| | | Two sample labels have been inter-changed | Samples accepted by the system but reports and invoices are sent to wrong customers | High |
| | | Sample label is missing | Sample not accepted by the system and sample must be disposed | High |

Not all events are caused by software errors, as seen in the example above, but all conditions have to be taken into account when assessing the risk.

## Risk Assessment

A conventional way to assess the risk is to classify it and then assign it a priority.



**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

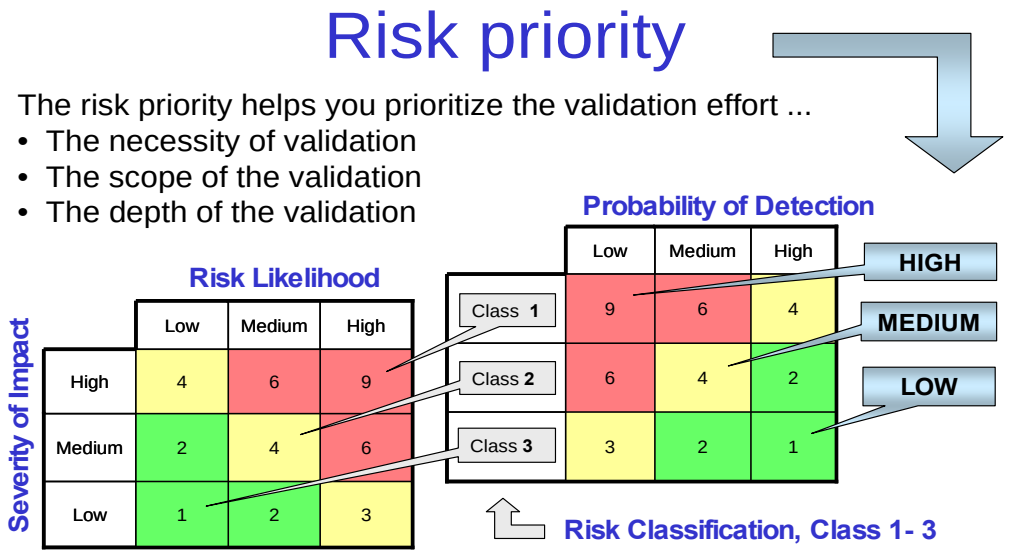
When using this model, the next stage after the event has been identified, is to assess the severity of the impact and the likelihood for the event to occur. The likelihood is commonly thought of in terms of frequency or probability. The combination of severity and likelihood is said to classify the risk as shown in the example below.

| Risk Class | Severity of impact & Likelihood of occurring | Vulnerability |
|---|---|---|
| **Class 1** | Very severe impact,<br>Very likely to occur. | The risk is unacceptable and should to be eliminated at once. |
| **Class 2** | Moderate impact,<br>Reasonable likely to occur. | The risk is acceptable for now but should be taken into account and be eliminated as soon as possible. |
| **Class 3** | Minor negative impact,<br>Almost unlikely to occur. | The risk may be ignored but should be eliminated when convenient. |

Some risks pose a bigger threat than others, even if they belong to the same risk class. An error event which is very difficult to detect may persist for a long time, while if revealed at once, it can be corrected at once. E.g., a systematic software error that causes all calculated results to be 5% too low can be difficult to detect, while results consequently being out of range are far more obvious and will probably be detected at once. It may be a significant problem that all incorrect reports delivered until the error is detected must be recalled - and the longer that takes, the greater is the impact.

The probability of detection may commonly be thought of in terms of detected error events per number of transactions or operations.

By combining the risk classification with the probability of detection, it is possible to calculate a priority for the fault condition based upon the greatest vulnerability. Even if many potential risks may be eliminated by the software testing, an adequate validation will be required, and the risk priority is a useful measure of the need for risk-based validation.




The numbers in the tables may be used to obtain a detailed priority score based on the input estimates of severity of impact, the risk likelihood, and the probability of detection.

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

The calculation of the risk factor described in this document is more straightforward than shown above since the vulnerability expressed by the risk classification is not emphasized. The primary outcome of the risk assessment is a priority score that indicates the need for risk-based validation, as shown simplified in the table below.

| Risk Priority | Validation | Example |
|---|---|---|
| **High** | Validation is definitely required at once since we are dealing with a serious error event in a vulnerable process<br>- may cause severe impact<br>- is very likely to occur<br>- may be difficult to detect | There may be a problem with the operating instructions since it has been observed that different persons get different analysis results.<br>**Note 1** |
| **Medium** | Validation should be performed as soon as possible since we are dealing with an adverse event in a common process<br>- may cause moderate impact<br>- is reasonably likely to occur<br>- will probably be detected rather quickly | There may be a problem with the database since it has been registered that distinct analyses have been carried out with the wrong reference data.<br>**Note 2** |
| **Low** | Validation should be performed when convenient since we are dealing with an error event in a rather resistant process<br>- will cause insignificant impact<br>- is almost unlikely to occur<br>- is detected immediately | There may be a problem with the software printing labels since artificial test sample identities are sometimes invalid and abort the analysis.<br>**Note 3** |

**Notes**

1. Personnel that operate the software associated with a process pose a potential risk and it is therefore essential that the personnel are well educated and that the user manuals are understandable and approved by means of peer review. Many laboratories control their work by written procedures and instructions and this information must be validated as well.
2. Validation must also include the operating phase in which all adverse and unexpected events should be registered in a logbook. Acceptable anomalies and malfunctions will normally be bypassed by workarounds and precautionary steps until finally repaired.
3. A low risk priority should never be used as an excuse for omitting validation. All software in all processes must be proper validated - the outstanding question is only when and how much. In addition, an identified event with even low risk priority may cover up a far more risky scenario that could be revealed by the validation.

⇩ Table of Contents

## Risk-based Validation

Validation may be carried out in accordance with the Nordtest Software Validation method which recommends the use of a life cycle model with 6 phases. If the risk assessment approach is included in each of these phases, it may be referred to as a risk-based validation.

Adverse events caused by human errors have to be included in the risk assessment process, and a risk-based validation should also take areas such as education and authorisation of personnel, working routines, substitutes for absent operators, etc. into account.

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

## 1. Requirements and system acceptance test specification

The requirements should describe and specify the computer system completely and form the basis of the development and validation process. A set of requirements can always be specified. In case of retrospective validation (where the development phase is less relevant) it can at least be specified what the system is purported to do based on actual and historical facts. The requirements should encompass everything concerning the use of the system.

For each relevant item in the requirements and the system acceptance test specification, possible error events should be identified and their severity be assessed according to the risk assessment method outlined in this document. E.g. a possible error event could be that a certain function does not work exactly as specified and the corresponding system acceptance test does not catch the malfunction because it only occurs under rare conditions.

The risk-based approach may also help in specifying requirements for errors and alarms, especially if the vulnerability is assessed in a system perspective.

## 2. Design and implementation process

The design and implementation process is relevant when developing new systems and when handling changes subjected to existing systems. The output from this life cycle phase should be a software program approved and accepted for inspection and testing in the subsequent phase.

The design input phase establishes that the requirements can be implemented. Requirements that are incomplete, ambiguous, conflicting, or pose a significant risk are resolved with those responsible for imposing these requirements.

The design section serves as an entry for all changes applied to the computer system, also computer systems being subjected to retrospective validation. Minor corrections, updates, and enhancements that do not impact other modules of the system are changes which will not require an entire revalidation. Major changes are reviewed in order to decide the degree of necessary revalidation or update of the requirements and/or the system acceptance test specification.

All changes applied and all anomalies found and circumvented in the design and implementation process should be subject to a risk assessment in order to decide if the design has to be changed or the risk can be accepted. Quality cannot be tested into software; it must be included in the design from the outset.

## 3. Inspection and testing

The inspection and testing of the computer system should be planned and documented in a test plan. The extent of the testing should be in compliance with requirements, system acceptance test specification, purpose, complexity, risks, and the intended and expected use of the computer system.

The test plan should be created during the development or reverse engineering phase and should identify all elements that are about to be tested. The test plan should explicitly describe what to test, what to expect, and how to do the testing. Subsequently, it should be confirmed what was done, what was the result, and whether or not the result was approved.

If the preparation of a test plan for risk-based validation takes the risk assessment into consideration, the scope and level of testing may be increased for processes of great vulnerability and may correspondingly be decreased for processes with very low risk associated with the occurrence and consequences of fault conditions.

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

## 4. Precautions

When operating in a third-party software environment, such as Microsoft® Windows and Office, some undesirable, inappropriate, or anomalous operating conditions may exist. A discrepancy between the descriptions of the way an instrument should operate and the way it actually does may also be regarded as an anomaly. Minor errors in a computer system may sometimes be acceptable if they are documented and/or properly circumvented. Whatever you do, all precautionary steps taken should be subject to risk assessment.

## 5. Installation and system acceptance test

The validation of the installation process should ensure that all system elements are properly installed in the host system and that the user will obtain a safe and complete installation, especially when installing software with serious consequences.

After the installation, the system should be tested to an extent dependent upon the use of the system and the actual requirements. An adequate test should be specified in the validation test plan. Sometimes it is recommendable to carry out the installation testing in a copy of the true environment in order to protect original data from possible fatal errors due to using a new program. The outcome of such an installation testing may be used to identify critical events that can pose risks to the actual system.

The system acceptance test should be carried out in accordance with the system acceptance test specifications after installation. The computer system may subsequently be approved for use.

## 6. Performance, servicing, maintenance, and phase out

In this phase the computer system will be in use and is subject to requirements for service, maintenance, performance and support. This phase is where all activities reside during performance and where decisions about changes, upgrades, revalidation, and phase out are made.

All error events observed while using the computer system should be registered in a logbook, even if they seem harmless and are easy to handle. The process of risk assessment relies on the knowledge of possible error events (including those never imagined) that can pose any immediate or long-time threat to the computer system.

Validation of computer systems is a dynamic process which is pursued during the entire lifetime of the system and proper risk assessment should be performed at any time a change is to be applied to the system or a fault condition occur.

When a new version of the computer system is taken into use, the effect on the existing system should be carefully analyzed and the degree of revalidation decided. Special attention should be paid to the effect on old spreadsheets when upgrading the spreadsheet package.

It should be taken into consideration how (and when) to discontinue the use of the computer system. The potential impact on existing systems and data should always be examined prior to withdrawal. If phase out implies migration to another computer system, the risk scenarios identified in the old system can often be transferred directly to the new one.

## Procedures and Operating Instructions

Procedures and operating instructions are normally part of the company's quality system and are not directly included in the computer system validation. However, incomplete or misleading instructions

**Carl Erik Torp**
M.Sc. (Elec. Eng.)
Ejbydalsvej 239
DK-2600 Glostrup
Denmark

may pose a serious risk to the operation of the computer system and all relevant documents should therefore be included in the risk assessment process.

## Table of contents