Linear Algebra for QFD Combinators

A Tutorial for QFD Practitioners
how to Combine Measurements with Deployments

Dr. Thomas M. Fehlmann

Euro Project Office AG

Zeltweg 50

CH-8032 Zurich, Switzerland

Phone: +41 1 253 1306
Fax: +41 1 253 1364

E-mail: Thomas.Fehlmann@e-p-o.com

V1.1-7, 18 October, 2003

Linear Algebra for QFD Combinators

Dr. Thomas M. Fehlmann
Euro Project Office AG, Zurich, Switzerland

Summary

One of the most prominent tools in QFD is the matrix.

Matrices are well known in mathematics as a means to represent linear mappings between vector spaces. We use similar matrices to represent cause – effect correlations. Vectors, the elements of a vector space, represent Customer's Needs profile or a product profile. An interesting property of vectors is their linear independence. From this viewpoint we face two questions:

- What means linear independence for the profiles represented by vectors?
- When does the QFD matrix preserve linear independence?

QFD matrices are constructed from cause – effect relationships. Thus they represent a linear mapping from the solution space into the goal space. However, when calculating the solution weights, we use the matrix the other way round. Is this correct?

This paper gives answers both from a mathematical viewpoint and from practical experiences.

Thomas M. Fehlmann, Dr. sci. Math. ETH, Euro Project Office AG, Zeltweg 50, CH-8032 Zurich, phone +41 1 253 1306, fax +41 86 079 332 7056, E-mail: Thomas.Fehlmann@e-p-o.com

Table Of Contents

Table of Figures

1. Basics of Linear Algebra

1.1 Vector Spaces

We may assume the reader is familiar with the concept of a *vector space*. Everybody knows the physical 3 – dimensional Euclid space we are living within and is able to combine its vectors (straight paths) to traverse the space as suitable. Many people are familiar with the 4 – dimensional time – space vector space introduced by Albert Einstein to model relativistic effects[1]. Mathematicians naturally extend the number of dimensions to some finite number n and use it to model various theories and realities.

QFD practitioners use regularly n – dimensional spaces that consist of the dimensions of n customer's needs, and combine it with an m – dimensional solution characteristics space.

Vectors can be *added*: let $\underline{x}$, $\underline{y}$ be two vectors in a vector space. Then $\underline{x} + \underline{y}$ is a vector, and $\underline{x} + \underline{y} = \underline{y} + \underline{x}$.[2]

Moreover, there is a *scalar multiplication*. This scalar multiplication is linear: that means for each such scalar λ: $\lambda*(\underline{x} + \underline{y}) = \lambda*\underline{x} + \lambda*\underline{y}$. In Euclid space, the scalars are real numbers.

A set of vectors $\underline{x}_1, \ldots, \underline{x}_n$ is called *linearly dependent*, if there exist a nontrivial set of scalars $\lambda_1, \ldots, \lambda_n$ such that $\lambda_1*\underline{x}_1 + \ldots + \lambda_n*\underline{x}_n = \underline{0}$ (the zero vector) but $\lambda_i \neq 0$ for at least one index i.

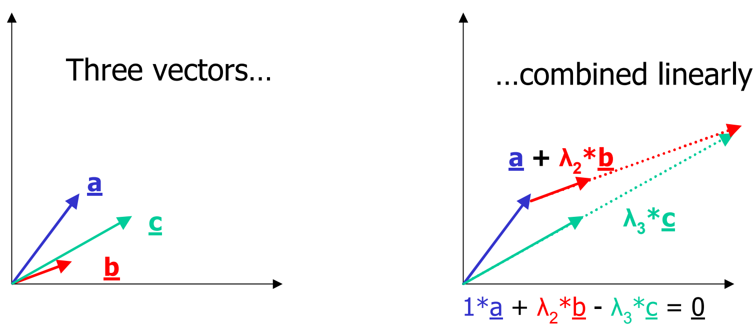

Figure 1: Set of Three Linearly Dependent Vectors in a Two - Dimensional Vector Space

A maximum set of vectors that is linearly independent is called a *basis* of the vector space. For instance, in a two – dimensional vector space every set of three or more vectors are linearly dependent.

1.2 Linear Mappings

Let A, B two vector spaces. A mapping φ: A→B is called *linear* if for all scalars λ: $\varphi(\lambda*\underline{x}) = \lambda*\varphi(\underline{x})$ holds.

[1] Note the time – space continuum is skew – symmetric. Some QFD practitioners use roofs for their matrices that serve a similar purpose as a skew – symmetric signature on vector spaces.

[2] QFD practitioners use addition but with caution: although the model allows it, adding product features is often more difficult in practice than in theory.

It is easy to see that the *image* of A in B can have at most the dimension of A. It can have fewer dimensions, including zero, if linear independent vectors in A are mapped to linear dependent images in B. However, increasing the dimensions is not possible.

1.3 Matrix Representation

Let $\underline{e}_1, \dots, \underline{e}_n$ be a base of A and $\underline{f}_1, \dots, \underline{f}_m$ a basis of B, n, m their dimensions, respectively. Then every vector $\underline{x}$ can be written as

$$\underline{x} = \lambda_1 * \underline{e}_1 + \dots + \lambda_n * \underline{e}_n.$$

Let φ be a linear mapping. Then

$$\varphi(\underline{x}) = \lambda_1 * \varphi(\underline{e}_1) + \dots + \lambda_n * \varphi(\underline{e}_n)$$

is a vector in B.

Because $\underline{f}_1, \dots, \underline{f}_m$ is a basis of B, every vector $\underline{y}$ of B can also be written as a linear combination

$$\underline{y} = \mu_1 * \underline{f}_1 + \dots + \mu_m * \underline{f}_m.$$

In particular,

$$\varphi(\underline{e}_i) = \mu_{i,1} * \underline{f}_1 + \dots + \mu_{i,m} * \underline{f}_m \text{ for } i = 1, \dots, n.$$

Thus the matrix coefficients $\mu_{i,j}$, where i = 1, …, n and j = 1, …, m define the mapping φ uniquely in terms of the bases $\underline{e}_1, \dots, \underline{e}_n$ and $\underline{f}_1, \dots, \underline{f}_m$. The coefficients $\mu_{i,j}$ form a matrix that represents φ.

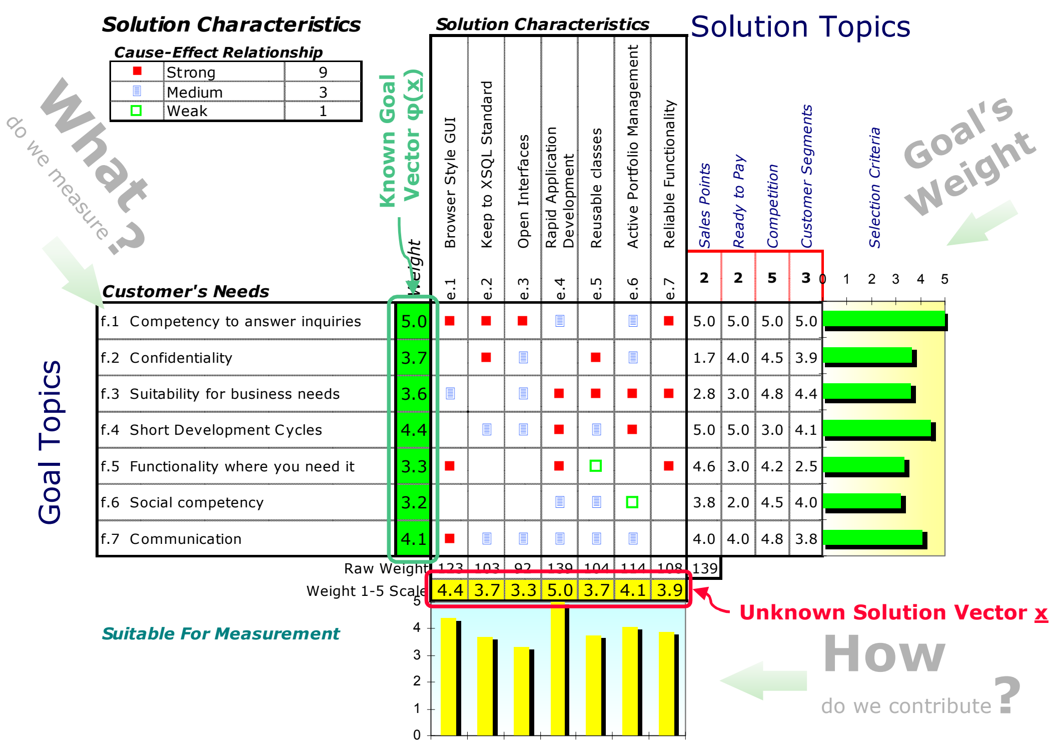

Figure 2: Sample Linear Mapping between Topic Spaces

For a QFD practitioner, this matrix is what he uses to describe the cause – effect between the causes $\underline{e}_1, \ldots, \underline{e}_n$ that form the topic space A and the effects $\underline{f}_1, \ldots, \underline{f}_m$. The dimension of the topic space is the minimum set of characteristic elements that he needs to describe a topic. The addition of topic elements corresponds to its combination, and the scalar multiplication is the weight of the topic element.

Note that when we build a QFD matrix we define the matrix coefficients as the impact of the solution characteristics on the goal topic. This is because QFD matrices are in fact a convenient way to denote cause – effect diagrams (Ishikava – diagrams). Thus the mapping $\varphi: A \rightarrow B$ describes the cause – effect relation from one topic A into another B. Its coefficients $\mu_{i,j}$ consist of the *strength* indications between the causes $\underline{e}_1, \ldots, \underline{e}_n$ and the effects $\underline{f}_1, \ldots, \underline{f}_m$.

1.4 Normalization

The direction of the vector is defined by the values of its coefficients relative to each other. We have seen that a vector can be multiplied with a scalar. This does not change the direction of the vector but affects only its length. We use that feature to normalize our vectors (the weights in QFD) to a range 0 … 5. This is done by the formula $\xi_i / \max_i(\xi_i)*5$.

The choice of 5 is purely arbitrary and could be any nontrivial number; most convenient would be 1. This is a common technique with vector spaces. It allows drawing nice profiles to compare different weights with each other.

A more stringent restriction is that our vectors use only four values for the coefficients: 0, 1, 3, 9. This restricts us to positive coefficients, and this is a crucial restriction.

There exist different flavors in QFD both for coefficients and scalars, however it still is possible to look at it as a kind of linear mapping between topics under some restrictions. We call such profiles *Metric Profiles,* or simply *Metrics*.

Customer's Needs

f.1	Competency to answer inquiries	5.0
f.2	Confidentiality	3.7
f.3	Suitability for business needs	3.6
f.4	Short Development Cycles	4.4
f.5	Functionality where you need it	3.3
f.6	Social competency	3.2
f.7	Communication	4.1

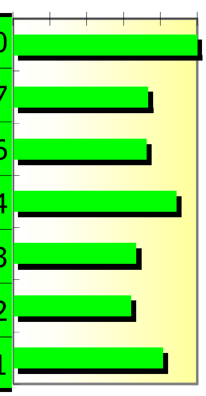

Figure 3: Sample Metric Profile for Customer's Needs

2. Hidden Properties of QFD

2.1 Dimensions

When we do QFD we start not with vectors but with topic such as customer's needs or solution characteristics. One of our standard concerns is that such topics tend to grow ad infinitum in the number of characteristics that we like considering. There exist techniques like VoC or AHP to limit growth of its size

However, linear mathematics also offers a very simple device. Topic characteristics that are dependent on each other behave like linear dependent vectors. When you create a QFD matrix and have two topic characteristics that both yield the same effect on the goal characteristics, the matrix will contain two identical columns. In general this may happen as well if you have a couple of solution characteristics that are dependent on each other. Exactly this is the behavior of linear dependent vectors in a mapping.

Detecting linear dependency in a matrix is a standard task in linear algebra. The simplest means are the *determinants*. The determinant of an n x n square matrix is zero exactly if it contains linear dependent rows or columns.

The determinant is calculated by the sum of all products between each permutation of pairs of matrix coefficients, with an alternating sign[3].

$$\Sigma_{\,i,j,k}\,(-1)^{\sigma(i,k)} * \mu_{i,j} * \mu_{j,k}$$

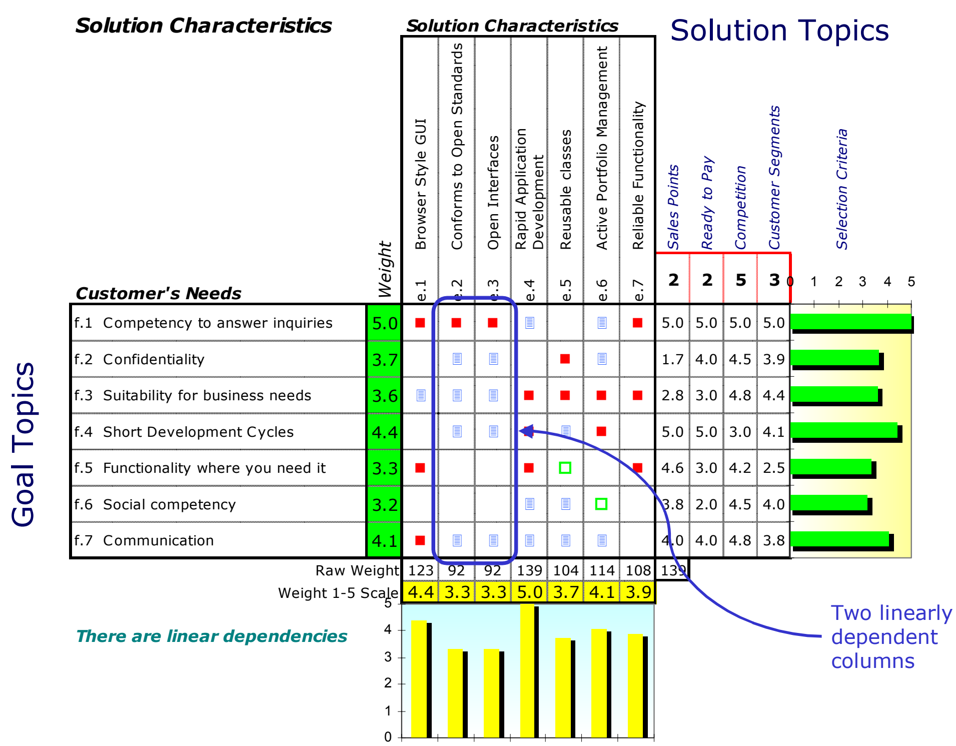

Figure 4: Sample Matrix with Linear Dependent Columns

[3] In Excel, it can be computed by the build-in function MDETEREM

Thus the determinant may be used to detect linear dependency within topic characteristics. Indeed, whenever there exists an n x n sub–matrix, where n is the minimum of the two topic dimensions n and m, whose determinant is zero, then there must be a linear dependency either within the solution topics or the goal topics.

The figure above shows a very simple case of two linear dependent columns. By changing the solution characteristics slightly, both now refer to the same solution characteristic and thus have exactly the same impact on customer's needs. Such characteristics should be merged together whenever possible.

In reality, if you have for instance solution topics consisting of 70 features that you like to combine with the process characteristics to get the features' importance relative to each other in view of the customer's needs, then such redundancies are pretty hard to see. Computing the determinant is a simple tool for checking for the presence of such

2.2 Using the Matrix

From this construction method, the coefficients $\mu_{i,j}$ of the matrix are defined such that whenever $\underline{x} = \langle \xi_1, \ldots, \xi_n \rangle$ is a weight for the solution topics, the corresponding weights for the goal topics is

$$\varphi(\underline{x}) = \langle \Sigma_{i=1\ldots n} \mu_{i,1} * \xi_i, \ldots, \Sigma_{i=1\ldots n} \mu_{i,m} * \xi_i \rangle.$$

The vector $\underline{x} = \langle \xi_1, \ldots, \xi_n \rangle$ are the unknowns. From the resulting vector $\varphi(\underline{x}) = \langle \zeta_1, \ldots, \zeta_m \rangle$. we know the coefficients ζ_i from a suitable measurement. For instance in the House Of Quality well established measurements are available thanks to Voice Of The Customer (VoC) and Analytical Hierarchical Process (AHP) for the Customer's Needs $\varphi(\underline{x}) = \langle \zeta_1, \ldots, \zeta_m \rangle$. We want to know what weight the solution topics should get in order to yield the weights of customer's needs as exactly as possible. The better we can meet our solution topics profile, the less costly it is for the supplier to exactly fulfill customer's needs.

From a mathematical viewpoint, this means that we have to compute the solution of an over –defined linear equation system defined by the cause – effect combinator matrix. This system has the form:

$$\zeta_1 = \Sigma_{i=1\ldots n} \mu_{i,1} * \xi_i$$

$$\ldots\ldots\ldots\ldots$$

$$\zeta_m = \Sigma_{i=1\ldots n} \mu_{i,m} * \xi_i$$

It must be solved for the unknown coefficients ξ_i. An exact solution is only possible if $n \geq m$ and whenever the determinant of the maximum square sub-matrix is not zero.

If $n > m$ then more than one solution may exist. This is welcome for finding an optimum solution. Thus it comes that QFD matrices in a comprehensive development always grow in size[4].

There exist standard algorithms for solving such systems. You will find those algorithms in the textbooks for operations research and linear optimization. Most often used is the Simplex – Algorithm.

[4] This seems natural. But is there anything not "natural" in quality management, after all?

2.3 Why does QFD work?

Let $\mu_{i,j}$ denote the matrix that represent the linear mapping φ.

Let $\underline{x} = <\xi_1,\ldots,\xi_n>$ be the vector in the topic space A with the coefficients ξ_i and $\underline{y} = <\zeta_1,\ldots,\zeta_m>$ a vector in topic space B. Note that we can use the matrix in two directions:

(1) One to compute the vector
$\varphi(\underline{x}) = <\Sigma_{i=1\ldots n}\, \mu_{i,1}*\xi_i,\ \ldots,\ \Sigma_{i=1\ldots n}\, \mu_{i,m}*\xi_i>$

(2) Another to compute the vector
$\psi(\underline{y}) = <\Sigma_{j=1\ldots m}\, \mu_{1,j}*\zeta_j,\ \ldots,\ \Sigma_{j=1\ldots m}\, \mu_{n,j}*\zeta_j>$.

Obviously $\psi(\varphi(\underline{x})) \neq \underline{x}$ and $\varphi(\psi(\underline{y})) \neq \underline{y}$, i.e. the two linear mappings are not inverse to each other. However, in QFD practice we assume somehow that they are. We construct a matrix φ that represents the cause – effect impact of topic space A onto space B. Then we take the weight vector $\underline{y}$ in the goal topic space B and use ψ: B → A to get the weight vector $\psi(\underline{y})$ ε A for the topics in the solution space A.

In general there is no unique solution $\underline{x} = \varphi^{-1}(\underline{y})$. If such a unique solution exists, it is against the idea of finding an optimum solution for the solution topics.

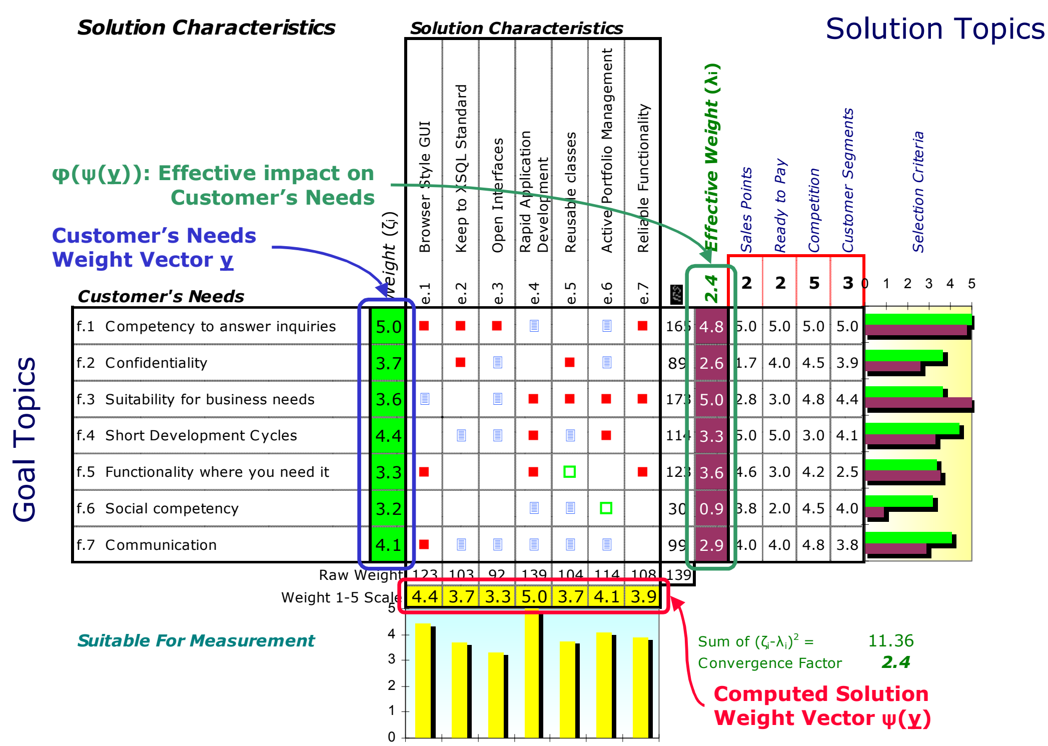

Figure 5: QFD Cause – Effect Combinator with Bad Convergence Factor

The approach used in workshops and on paper, namely to use the pseudo – inverse function ψ to compute the weights of the solution topics, works at least as a first approximation. It does not necessarily yield the optimum result. It yields a profile metrics that looks similar to the one sought, so we use that instead of doing linear optimization.

2.4 A Closer Look

The above sample shows two profiles in the graph to the left of the matrix. The original customer's needs metric profile

$$\underline{y} = <\zeta_1, \ldots, \zeta_7> = <5.0, 3.7, \ldots 4.1>$$

is shown in light green in the profile graph. We expect it to be the result of some VoC analysis and thus well established.

Using the pseudo – inverse mapping ψ, we get the computed solution weight vector $\psi(\underline{y})$ as usual. This yields the solution profile at the bottom; see (2).

Now we compute the effective impact of $\psi(\underline{y})$ on Customer's Needs, using (1):

$$\varphi(\psi(\underline{y})) = <\lambda_1, \ldots, \lambda_7> = <4.8, 2.6, \ldots, 2.9>$$

This is again a vector in the same space as the original customer's needs metric profile y, however, it is significantly different from y. In the profile graph to the right of the matrix, this second profile turns up in dark brown.

Since the matrix has been defined as a cause–effect matrix (originally, as a sequence of Ishikava – Diagrams), this weight vector φ(ψ(y)) = <λ1, …, λ7> is indeed the customer value we really get when choosing the solution weight vector ψ(y).

There is an important gap between $\underline{y}$ and $\varphi(\psi(\underline{y}))$. The gap tells us something about in what respect our choice of solution topics cannot really match the needs of the customer. Such gaps do always exist except in an ideal world[5].

Some matrices have big gaps and some rather small gaps. We can narrow the gap by choosing better solution characteristics. Thus it is tempting to define a metric to measure the quality of our choice of solution components. In our sample case, we got a Convergence Factor of 2.4.

2.5 The Convergence Factor

In QFD practise, you easily run into problems when you define a matrix that needs heavy optimization, because its first approximation does not yield satisfactory results. Thus it was found useful to define a metric, how well the measurements match the predictions from the deployment. This metric we call the *Convergence Factor*.

The formula for the Convergence Factor is the length of the vector difference between measured weight and derived weight, divided by the number of profile coefficients. Let $\underline{y} = <\zeta_1, \ldots, \zeta_n>$ and $\underline{x} = <\lambda_1, \ldots, \lambda_n>$ be two such vectors. Then the corresponding Convergence Factor is the length of its vector difference $\underline{y} - \underline{x}$ divided by the number of coefficients n, times 5[6]:

$$5 * \left(\Sigma_{i=1\ldots n} (\zeta_i - \lambda_i)^2 \right)^{1/2} / n$$

[5] Note that this sheds another light on how TRIZ and QFD are interrelated in each other.

[6] The factor 5 is only to avoid an unhandy threshold of 0.2. The Excel formula reads

SQRT(SUMXMY2(<$\zeta_1,\ldots,\zeta_n$>; <$\lambda_1,\ldots,\lambda_n$>)) / COUNT(<$\lambda_1,\ldots,\lambda_n$>)*5

where you have to replace the vectors (<$\zeta_1,\ldots,\zeta_n$> and <$\lambda_1,\ldots,\lambda_n$> by their respective Excel ranges, and the COUNT – function simply returns n, the number of coefficients defined in that range.

A Convergence Factor of zero means complete convergence; up to one it is considered acceptable. Convergence Factors greater than one indicate a significant difference between the deployed weight and the measured weight of the topic profile.

2.6 Optimization

There are two solution approaches

1. A QFD practitioners solution: Add another solution component that better supports Customer's Need f.6: "Social competency".
2. A mathematical solution using the Simplex algorithm for getting an optimum solution

The big gap in matrix "Figure 5: QFD Cause – Effect Combinator with Bad Convergence Factor" is not a big surprise when inspecting the matrix more closely. We then see that the Customer's Need f.6: "Social competency" gets almost no support by the chosen solution components e.1, …, e.7. Thus when we try to optimize $\psi(\underline{y})$ using a Simplex algorithm we find it difficult to get better weights $\underline{x}_{better}$ for the solution topics even if we are able to somewhat narrow the gap between $\underline{y}$ and $\varphi(\underline{x}_{better})$.

It seems there is no generally applicable optimization method to do the optimization even if we can identify such situations where optimization is possible. This is shown by the following example:

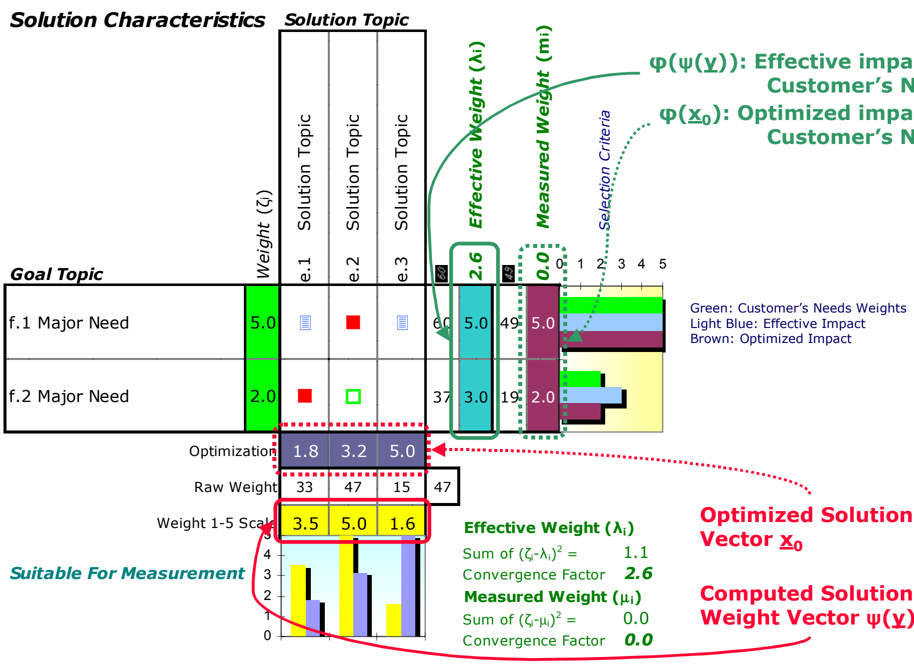

Figure 6: QFD Cause – Effect Combinator with Optimized Solution Weight

As we can see, the pseudo – inverse function ψ yields a result that does not model the fact that customers graded the "Major Need" f.1 much higher than f.2. A better approach is to increase the relative weight of e.3 even if its effect on the customer's need f.1 is just medium. If e.3 does not cost too much, this might be the better solution. The optimized solution has nothing to do with the computed weight scale using formula (1).

This example suggests that we need to investigate further into how to make linear optimization available to QFD practitioners.

2.7 Finding a Better Solution Vector

However, such a mathematical approach has the big disadvantage that its mechanics remain cryptically incomprehensible to most average QFD users and even practitioners. Moreover, it will not always return reasonable results.

Thus in most cases the much better approach is to review the chosen set of solution components. In our sample, we already noted that Customer's Need f.6: "Social competency" gets almost no support. Therefore we add e.8: "Moderated Discussion Forum" to the solution components; and we replace e.4: "Rapid Application Development" by "Agile Programming"; assuming the latter will add more to f.6: "Social competency".

This is the resulting House of Quality:

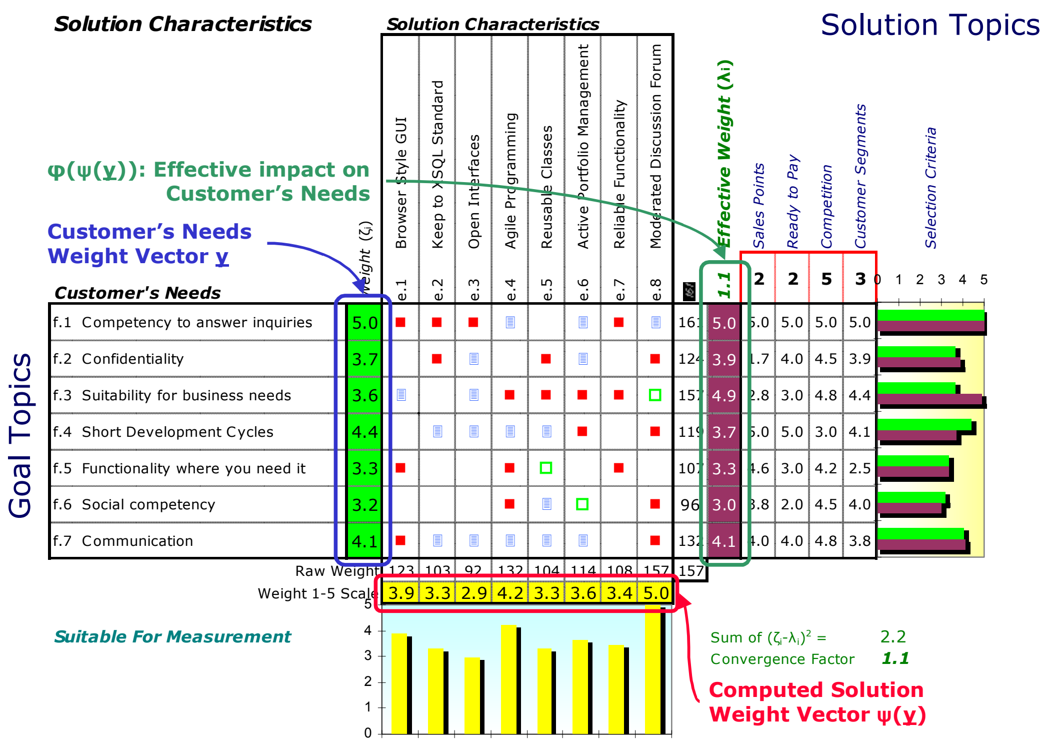

Figure 7: QFD Cause – Effect Combinator with Improved Choice of Solution Components

Now we have a much-improved Convergence Factor of 1.1; the only fault is that we overachieve in terms of f.3: "Suitability for business needs". Possibly we can live with that[7].

[7] It may sound strange, but it is important ***not*** to overachieve. QFD is about optimizing resources in order to get the most out of it, not about doing everything that sounds good. For the latter, a simple gap analysis will do. The New Lanchester theory defines the metric to tell us how much is enough.

It is obvious that the revised set of solution approaches is much better than the original one. The Convergence Factor seems a good metric for that.

2.8 Consequences

We draw the following conclusions. When setting up a matrix, a few checks are necessary:

- The matrix must not contain linear dependent vectors. This can be checked with the determinant check.
- The calculation of weights for the causing topics by formula (1) is not reliable without check.
- The Convergence Factor seems a good check to validate your matrix.
- Before trying mathematical optimization, try a better solution approach

We discourage approaches with negative matrix coefficients. They may yield completely wrong results.

3. Combinations

3.1 QFD Combinators with Measurements

We call a cause – effect matrix a *QFD Combinator*, if it is used in both directions. Note that this does not mean that the linear mapping is invertible.

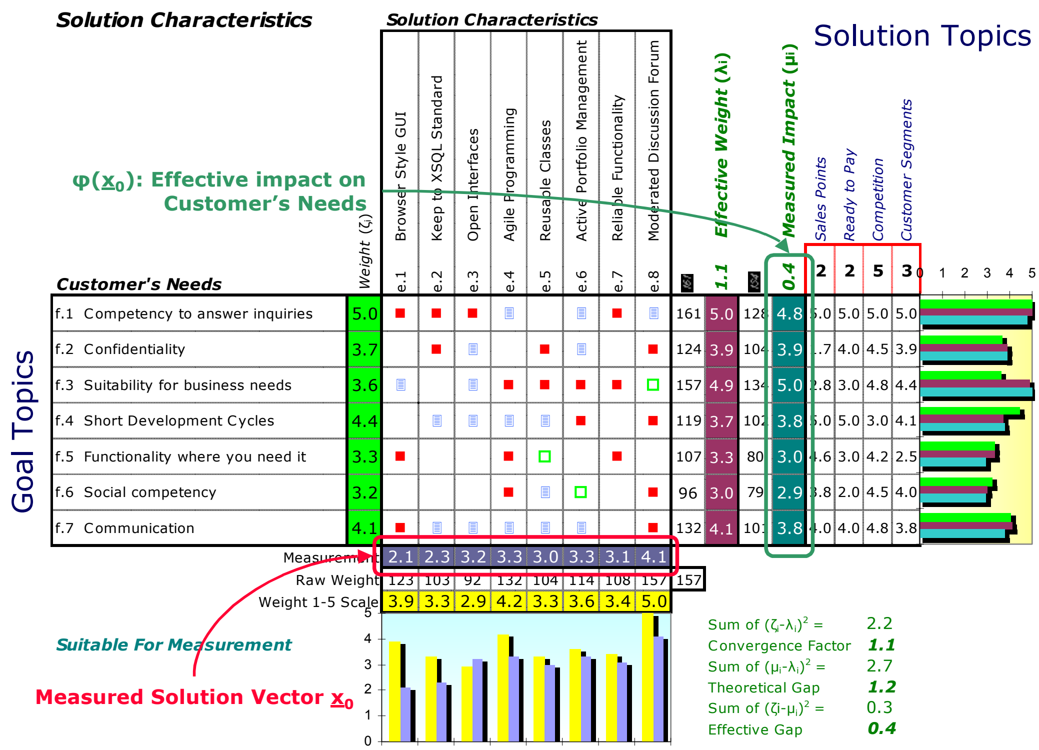

Figure 8: QFD Cause – Effect Combinator with Measurement

Main applications of matrices combining deployment and measurement are with Risk Management, Testing, and New Lanchester Theory.

Note that the above sample matrix expects a measurement for the solution topics, in this case for the solution requirements. Usually solution requirements cannot be measured directly. However, you may want to compare existing prioritizations with a QFD deployment based on customer's needs that may have been established with a Voice of the Customer process.

In general, this technique allows constructing a metrics nets based on QFD in the Broad Sense. We can deploy forward and compare results backward through a network of QFD combinators. The gaps that can be observed with such a technique are most valuable information not only for managing a business or an organization, but also for strategic leadership.

3.2 Combining Linear Mappings

Given two deployments φ_1 and φ_2. Clearly $\varphi_1 \circ \varphi_2(\underline{x}) = \varphi_2(\varphi_1(\underline{x}))$ defines the composition of two linear mappings. When we carry that composition over to matrices, we can see that the combination of two combinators simply is represented by the matrix multiplication, thus by the matrix $(\Sigma_j \mu_{i,j} * \nu_{j,k})_{i,k}$.

Thus Comprehensive QFD (or QFD in the Broad Sense) is quite nicely modeled and understood by Linear Algebra.

3.3 Transitivity

We have learned how to compute $A \circ B = C$, where A is the matrix representing φ and B for ψ. The question is: When we ask the same team who set up the matrices A and B to construct a direct deployment C' directly, will they come up with the same result as when they first do the matrix A then B and then compute $A \circ B$? In other words: Is C = C'?

The answer is certainly "No". We know from Prof. Akao that it is much better to compute A then B than to try to derive C' directly. Every causal correlation in C' needs then an argument involving some topics from B, and these arguments are then not well documented.

3.4 Combining Deployments and Measurements

In view of the optimization problem we presented before, QFD matrices should always used with the feedback function φ for checking the correctness. However, if we can deploy into some measurable topics, as listed in the last chapter of this paper, we have an additional means to test our assumptions and matrices.

If our measurement goes way off the deployment, we must investigate whether the matrices were wrong or our solution does not fulfill requirements.

3.5 Combinatory Metrics

We present here a sample metrics network for GMC Software AG, a Swiss software company in the personalized printing business.

This following picture shows the two most important deployments in software development. One is via the quality solution attributes and may end in the software development processes, for instance in the key process areas according the CMM-I model of the SEI [13].

The other is the traditional product deployment that starts with business processes and goes step by step into software design and implementation. It continues with the testing phases, which are, for simplicity, only shown by one representative example in this picture, namely feature testing [9].

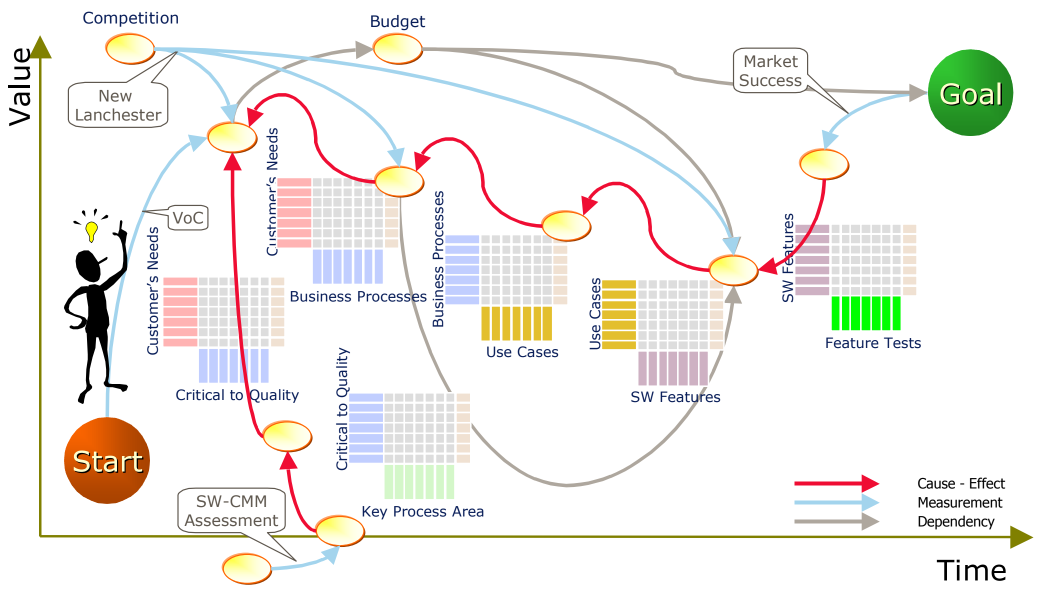

Figure 9: Sample Metrics Network for a Software Organization

4. Case Studies

4.1 New Lanchester

New Lanchester has been used in establishing the priorities for the product improvement project of former Swissair in 1999. This work is documented in the Proceedings of the 12th Symposium On Quality Function Deployment; June 2000 Novi, MI [5]. The measurements did compare well with the predictions by the deployments.

We used New Lanchester based market data to compare feature deployment with market data for GMC Software AG[8]. Again it turned out that the predictions match results. Moreover, we compared market data with the results of the ongoing Voice of The Customer process. Analyzing the gaps did show us how to make the business results rocketing. And this happened.

4.2 CMM Assessments

Measurements (i.e. a CMM Assessment) were used for improving software development processes in a Czech software house of GMC Software AG. This is documented in the Proceedings of the 8th International QFD Conference, September 2002, Munich, Germany [10].

8 This case study will be extensively discussed at the upcoming Software Measurement European Forum January 28 – 30 2004 in Rome, Italy.

Here we found that there were important gaps between what the market expects from the company and its actual state. This has been successfully addressed since.

4.3 Risk Assessments

There exist various views on risks. We compared the marketing view on a sample risk catalogue with the technical view. It did yield interesting insights in why agreement on risk assessment is sometimes difficult to achieve.

The work has been discussed and presented on the 4th European Conference on Software Measurement and ICT Control, Heidelberg, Germany, May 2001 [7].

Here φ and ψ were used to compare and match the two different views on risks. One measured topic are the Customer's Needs that deploy into risk threads. The higher the impact on Customer's Needs is the more important the risk.

The other view is the probability and impact evaluation of the risks which yields a different profile. We promote the use of the first view in risk management as well as the second.

4.4 Test Coverage

Finally, testing clearly should be done based on priorities deployed out of customer's needs. This compares with the measured test coverage. In this case the test coverage measurement is a means to assess the validity of the selected test cases for the purpose.

More on such test cases can be found in [9].

4.5 Strategy Deployment

In strategy deployment, measurements are the key to using Comprehensive QFD to lead businesses to success. Comparing deployments with actual business numbers allows optimizing the business strategy.

More on this application of the feedback function φ is published in [11].

References

[1] Yoji Akao et.al.: Quality Function Deployment (QFD); Productivity Press 1990

[2] Yoji Akao, QFD and the international stamdard ISO 9001, 7th International QFD Conference, October 2001, Tokyo, Japan

[3] Erwin Engeler, The Combinatory Programme, Birkhäuser 1995

[4] Thomas Fehlmann, Quality Function Deployment for the Full Business Life Cycle, EOQ Proceedings Vienna, April 1999.

[5] Thomas Fehlmann, Measuring Competitiveness in Service Design, in: QFD Institute (Ed.): 12th Symposium On Quality Function Deployment; June 2000 Novi, MI

[6] Thomas Fehlmann, Christian Hauri, Measuring Project Management Excellence, in: 3rd European Conference on Software Measurement and ICT Control, FESMA – AEMES, Oct. 2000, Madrid, Spain

[7] Thomas Fehlmann, Risk Exposure Measurements on Web Sites, in: 4th European Conference on Software Measurement and ICT Control, FESMA – DASMA, May 2001, Heidelberg, Germany

[8] Thomas Fehlmann, QFD as Algebra of Combinators, 7th International QFD Conference, October 2001, Tokyo, Japan

[9] Thomas Fehlmann, Business–oriented testing in E-Commerce, in: Software Quality and Software Testing in Internet Times, ed. Dirk Meyerhoff, SQS AG, Köln 2001

[10] Thomas Fehlmann, Combinatory Metrics for Software Development, 8th International QFD Conference, September 2002, Munich, Germany

[11] Thomas Fehlmann, Strategic Management by Business Metrics: An Application of Combinatory Metrics, International Journal of Quality & Reliability Management, Vol. 20 No. 1, 2003, Emerald, Bradford UK.

[12] Georg Herzwurm, Sixten Schockert, Werner Mellis: Qualitätssoftware durch Kunden-orientierung. Die Methode Quality Function Deplyoment (QFD). Grundlagen, Praxis-leitfaden, SAP R/3 Fallbeispiel. Vieweg-Verlag Braunschweig – Wiesbaden 1997

[13] Managing the Software Process, Watt S. Humphrey, Addison-Wesley, 1989

[14] Norikai Kano et.al., Attractive Quality and Must-be Quality, 12th Annual Meeting of the Japanese Society of Quality Control, 1982

[15] Glenn Mazur, QFD for Service Organizations, by Japan Business Consultants, Ltd., 1993

[16] IFPUG 4.1.1, International Funtion Point User Group, Function Point Counting Practices Manual, Release 4.1.1, Princeton Junction, NJ, April 2000

[17] Shigeru Mizuno, ed. 1988; Management for Quality Improvement: The 7 New QC Tools, Productivity Press, 1988

[18] Shigeru Mizuno and Yoji Akao, ed. 1994; QFD: The Customer-Driven Approach to Quality Planning and Deployment, translated by Glenn Mazur, Tokyo: Asian Productivity Organization, 1994

[19] Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, and Charles V. Weber, "Capability Maturity Model for Software, Version 1.1", Software Engineering Institute, CMU/SEI-93-TR-24, DTIC Number ADA263403, February 1993

[20] Nobuo Taoka: Lanchester Strategy – An Introduction, Lanchester Press Inc, 1997

[21] Ernest Wallmüller: Software – Qualitätsmanagement in der Praxis, Carl Hanser Verlag, May 2001

[22] The apparatus of Antikyra. Technical Musemum at Thessaloniki, exposed in: World Exhibition Hannover 2000

[23] Richard E. Zultner: Quality Function Deployment (QFD) for Software: Structured Requirements Exploration. In:. Schulmeyer/McManus (ed.): Handbook of Software Quality Assurance.2nd Ed., Zurich 1992, S. 297-319

[24] Richard E. Zultner: Blitz QFD: Better, Faster, and Cheaper Forms of QFD. In: American Programmer. October 1995, S. 25-36