Six-dimensional model of software usability

Georgy Saveliev
Auriga, Inc.
Moscow, Russian Federation
GeorgySaveliev@gmail.com

In an effort to address the need for a structured practical approach to building usable software, this paper proposes a CAUSER six-dimensional model of usability. The CAUSER abbreviation here stands for Comprehensibility, Attractiveness, Utility, Safety, Efficiency and Responsiveness. These six dimensions are considered as orthogonal axes, used to define overall usability of a software artifact as a vector in the Cartesian space.

This paper approaches measuring usability with a concept of reducing a level of uncertainty. It suggests using a ten-point normalized scale for capturing usability measurements and elaborates on proper application of statistical methods for analyzing them. It also gives an example of applying the CAUSER model for structuring informal usability knowledge, represented in a form of disconnected sets of rules and principles.

Keywords: usability, usability design, usability measurement, usability assessment, multi-dimensional model, level of uncertainty

Пытаясь удовлетворить потребность в применении структурированного практического подхода к созданию удобного программного обеспечения, данная статья предлагает шестимерную модель практичности программного обеспечения CAUSER. Аббревиатура CAUSER расшифровывается как Понятность (Comprehensibility), Привлекательность (Attractiveness), Полезность (Utility), Безопасность (Safety), Эффективность (Efficiency) and Отзывчивость (Responsiveness). Эти шесть измерений рассматриваются в качестве ортогональных осей, используемых для определения меры практичности программного артефакта в виде вектора в декартовом пространстве.

Данная статья подходит к измерению практичности как уменьшению уровня неопределенности. В статье предлагается использовать для измерения десятибальную нормализованную шкалу и обсуждаются вопросы правильного применения статистических методов для анализа результатов измерений. В статье также приводится пример использования модели CAUSER для структурирования неформальных знаний в области практичности программного обеспечения, представленных в виде разрозненных наборов правил и принципов.

Ключевые слова: практичность, удобство, программное обеспечение, измерение практичности, многомерная модель, оценка практичности, уровень неопределенности

I. INTRODUCTION

It is barely possible now to find out a true origin of the statement: “Nothing is more practical than a good theory”, repeatedly cited at different times by prominent scientists of various nations. I love it myself, but I also know many more pragmatic people who aren’t with me on this. However, if we substitute here “theory” with “model”, the aphorism will probably make more sense even for those of us who prefer another proverb: “The difference between “theory” and “practice” is that in theory there is no difference between theory and practice, but in practice, there is”.

The most evident practical result of developments in usability over last 20 years is an introduction of numerous “style guides”, claiming to be industry, platform or global standards. Essentially, they are collections of artifacts and heuristic rules, suggested as solution patterns for designing user interfaces. And, as often happens, a solution of a problem is becoming a problem in itself. The new problem is not only in choosing an “applicable standard” from the available manifold, but is also in its application. On one hand, it turns out that standard solutions do not cover versatility of real-life tasks faced by software designers. On the other hand, they don’t suggest a solid foundation for making decisions in situations where deviating from a standard is well justified.

A primary purpose of standards and patterns is preventing creation of apparently bad user interfaces. Following the standards helps us in producing software applications of a decent quality, but no more than that. The best products are not created by adhering to standards. On the contrary, bright design ideas, put into a foundation of innovative products, are becoming in future a part of one or another standard.

II. WHY WE NEED A MODEL?

Building perfect software requires accurate understanding of goals, rationale and results of choosing a particular design solution rather than merely following guidelines. Such understanding shall be based on a clear and functional set of concepts which one can use while designing and evaluating user experience.

We, humans, run on models. Whether we want it or not, whole our life is building and applying cognitive and behavioral models. And, of course, we imply and apply some model of a user experience when we create our software artifacts. We need good models for

- Structuring and articulating our formal and informal knowledge;
- Justifying research and development investments;
- Setting goals and measuring our progress towards them;

- Comparing designs objectively and measurably;
- Prioritizing and focusing our development efforts;
- Structuring and planning our testing activities.

The above may seem to be trivial, but it turns out that at the moment we have no explicit model of usability, and, ironically, even no thorough definition of the term "usability".

Two international standards define usability and human-centered (or user-centered) design:

- "[Usability refers to] the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." - ISO 9241-11
- "Human-centered design is characterized by: the active involvement of users and a clear understanding of user and task requirements; an appropriate allocation of function between users and technology; the iteration of design solutions; multi-disciplinary design." - ISO 13407

Based on ISO definition, Usability Body Of Knowledge project defines usability as "the degree to which something - software, hardware or anything else - is easy to use and a good fit for the people who use it" (www.uasbilitybok.org).

These definitions, however, do not clarify what's a "good fit", "satisfaction" or "easy to use". They also don't include often mentioned in professional literature concept of "learnability", which, at the end of 1990s replaced a widespread then concept of "intuitive" interfaces [1]. It might be considered as an attribute of "easy to use", but it isn't. There are many examples of systems which are easy to learn, but difficult to use and vice versa. Yet another important characteristic, which worth to be mentioned separately, is "responsiveness" [7].

The same is true about "satisfaction" – we may have a very different understanding of it. Donald Norman, the guru of usability, in his book "Emotional Design" [3], is examining the role of a user's emotional response and suggesting an idea that the ultimate indicator of software usability is how attractive it is for end users. He states it simply and clearly: "Attractive things work better". Kano comes with the "excitement" concept [9]. Whilst Eric Shaffer clearly puts his PET (Persuasion, Emotion, Trust) approach outside of the usability scope by prefacing it with "Usability is no longer enough" statement [5].

These vague terms are usually explained and clarified via many widespread rules and principles of design. The principles are not enough, though. As we know, principles and rules are things to be cleverly applied or violated when necessary. However they rarely explain what is a clever application or what we are trading off in terms of usability when violating a particular principle.

The Usability BOK states: "A foolish consistency...: There are times when it makes sense to bend or violate some of the principles or guidelines, but make sure that the violation is intentional and appropriate". It also says: "...experience should guide how those tradeoffs are weighed". But for weighting the tradeoffs objectively we need something more robust than a raw "experience" – we need a model.

Another problem with principles is that they might make much sense when considered in isolation. However, being put together, they don't form a coherent picture and are looking very much like Picasso's Three Musicians – you can recognize many familiar meaningful parts, but you cannot either properly bind or separate them.

III. Why a Multi-Dimensional Model?

The concept of Cartesian multi-dimensional space provides us a very useful decomposition, possessing two important qualities:

1. Ability to identify absolute and relative position of an artifact on each of the dimensional axes.
2. Ability to "move" the artifact along one axis without changing its position on the other axes.

These two abilities allow us reducing cognitive complexity by focusing on specific qualities in isolation from the others. On the other hand we can structure our knowledge by relating facts, measures, rules, tasks and methods to corresponding dimensions. The model is thus serving us as a framework for both building our subject area knowledge and applying it for designing artifacts. For example, car makers, in fact, are using a multi-dimensional model to match a specific market segment by varying rather independently wheel bases, transmissions, engines, interiors, exteriors and auxiliary functions of their vehicles.

Applying this approach to usability we can "measure" absolute and relative "positions" of design artifacts in each dimension. Then, once we are able to identify a "position" of an artifact on every axis, we can define its overall usability as a vector in the multi-dimensional space.

IV. CAUSER: Six Dimensions of Usability

For a model of usability, I'm suggesting six dimensions, abbreviated as CAUSER. The abbreviation stands for: Comprehensibility, Attractiveness, Utility, Safety, Efficiency and Responsiveness, which can be defined as follows.

Comprehensibility – ability of a user to understand meaning and usage of a system and its elements, relying on user's purpose and previous experience. It is also an ability of a user to understand a sequence of actions required for producing desired results. This understanding can be achieved by recognizing familiar elements and patterns, by exploring system's behavior in response to user's actions, by examining available hints and documentation, by consulting with other users or through a formal study of the system.

Attractiveness – ability of a system to engage a user emotionally by suggesting positive sensual (visual, audial, tactual) or cognitive experience.

Utility – ability of a system to assist users in executing their tasks for achieving desired results. The Utility attribute can manifest itself in expanding a range of achievable results, supporting their variations, improving their quality, increasing performance or otherwise extending user's capabilities. Utility is a measure of effectiveness in terms of ISO 9241-11 definition of usability.

Safety – ability of a system to recognize or prevent erroneous user actions, undo mistaken user actions, recover from errors and their consequences, guide a user through a safe path, assist in diagnosing and curing problems.

Efficiency – ability of a user to get desired results using minimum set of actions, natural for user's anatomy, psychophysiology and motor skills in the context of use. In particular this means an ability of a user to find and intake information without straining eyes and performing complex mental operations. Efficiency is measured as an inverse value of mental and physical energy spent to achieve a result. It includes "Ease of use" and can also be judged based on how many of required actions a user is able to automate so that they are performed subconsciously without distracting user's attention from executing a major task. The more obstacles and distracting factors a user meets on a way to desired result, the less efficient the system is.

Responsiveness – a timeliness and quality of feedback which a system provides about user actions, their actual and expected results, as well as changes of system's internal states.

V. Users and Contexts

Even though the dimensions of usability are well-defined, a position of an artifact on them is not an absolute, but a relative measuring. Why? Because a user and a context of use are defining together a "zero" point of the axes. As ISO definition states "…by specified users … in a specified context of use". Actually, we should separately measure the artifact's usability for different users in different contexts. For example, as we know, what youngsters adore, older people may regard as disgusting. Or a perfectly designed system with a touchscreen-based user interface becomes absolutely unusable for a person with busy or dirty hands. And so on.

Thus, from the CAUSER multi-dimensional model's point of view, users and contexts are not related to any particular dimension of usability. Instead, they define the origin of the usability coordinate system in its global outer space. Analysis and classification of users and contexts is a special area of research, involving anthropological, sociological, cultural and other aspects ([6], [8]).

VI. Using six dimensions for structuring usability knowledge

Now I want to proceed with an example of how the CAUSER model can be used for structuring usability knowledge presented in a form of heuristic rules and principles. Let's consider four different sets of them. The first is "34 Usability Maxims", published in 1997 by Arnold Lund. The second is Ben Shneiderman's "8 golden rules of Interface Design", published in 1998. The third is "Ten Usability Heuristics" by Jacob Nielsen (2004). And the fourth is the 20 "Principles of Usable Design" summarized by the Usability BOK project (2012) into 7 categories: Usefulness, Consistency, Simplicity, Communication, Error Prevention and Handling, Efficiency, and Workload Reduction. These four very different sets, when considered "as is", would make a reader wondering at how they are related to each other. However, being arranged by dimensions, they form six perfectly coherent sets. In the list below items of corresponding sets are prefixed with "L" (for Lund), "S" (for Shneiderman), "N" (for Nielsen) and U (for Usability Body of Knowledge project).

A. *User and Context*

L.1. Know thy user, and YOU are not thy user.

L.23. Design for regular people and the real world.

B. *Comprehensibility*

L.2. Things that look the same should act the same

L.5. Error messages should actually mean something to the user, and tell the user how to fix the problem.

L.8. Consistency, consistency, consistency.

L.10. Keep it simple.

L.19. Things that look different should act different.

L.20. You should always know how to find out what to do next.

L.22. Even experts are novices at some point. Provide help.

L.24. Keep it neat. Keep it organized.

L.28. Color is information.

L.29. Everything in its place, and a place for everything.

L.33. Let people shape the system to themselves, and paint it with their own personality.

L.34. To know the system is to love it.

S.1. Strive for Consistency.

N.10. Help and documentation. Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

N.2. Match between system and the real world. The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

N.4. Consistency and standards. Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

N.6. … Instructions for use of the system should be visible or easily retrievable whenever appropriate.

U. Relevance (Usefulness): The information and functions provided to the user should be relevant to the user's task and context.

U. Consistency and standards (Consistency): Follow appropriate standards/conventions for the platform and the suite of products. Within an application (or a suite of applications), make sure that actions, terminology, and commands are used consistently.

U. Real-world conventions (Consistency): Use commonly understood concepts, terms and metaphors, follow real-world conventions (when appropriate), and present information in a natural and logical order.

U. Simplicity (Simplicity): Reduce clutter and eliminate any unnecessary or irrelevant elements.

U. Self-evidence (Simplicity): Design a system to be usable without instruction by the appropriate target user of the system: if appropriate, by a member of the general public or by a user who has the appropriate subject-matter knowledge but no prior experience with the system. Display data in a manner that is clear and obvious to the appropriate user.

U. Structure (Communication): Use organization to reinforce meaning. Put related things together, and keep unrelated things separate.

U. Help and documentation (Communication): Ensure that any instructions are concise and focused on supporting the user's task.

C. Attractiveness

L.30. The user should be in a good mood when done.

L.32. Cute is not a good adjective for systems.

U. Supportive automation (Workload Reduction): Make the user's work … more fun.

D. Utility

L.14. The idea is to empower the user, not speed up the system.

L.18.The user should be able to do what the user wants to do.

U. Value (Usefulness): The system should provide necessary utilities and address the real needs of users.

E. Safety

L.3. Everyone makes mistakes, so every mistake should be fixable.

L.16. If I made an error, let me know about it before I get into REAL trouble.

L.21. Don't let people accidentally shoot themselves.

L.25. Provide a way to bail out and start over.

L.26. The fault is not in thyself, but in thy system.

L.31. If I made an error, at least let me finish my thought before I have to fix it.

S.5. Offer error prevention and simple error handling.

S.6. Permit easy reversal of actions.

N.3. User control and freedom. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

N.5. Error prevention. Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

N.9. Help users recognize, diagnose, and recover from errors. Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

U. Forgiveness (Error Prevention and Handling): Allow reasonable variations in input. Prevent the user from making serious errors whenever possible, and ask for user confirmation before allowing a potentially destructive action.

U. Error recovery (Error Prevention and Handling): Provide clear, plain-language messages to describe the problem and suggest a solution to help users recover from any errors.

U. Undo and redo (Error Prevention and Handling): Provide "emergency exits" to allow users to abandon an unwanted action. The ability to reverse actions relieves anxiety and encourages user exploration of unfamiliar options.

F. Efficiency

L.4. The information for the decision needs to be there when the decision is needed.

L.7. Don't overload the user's buffers.

L.9. Minimize the need for a mighty memory.

L.10. Keep it simple.

L.11. The more you do something, the easier it should be to do.

L.15. Eliminate unnecessary decisions, and illuminate the rest.

L.17. The best journey is the one with the fewest steps. Shorten the distance between the user and their goal.

L.27. If it is not needed, it's not needed.

L.28. Color is information.

S.2. Enable frequent users to use shortcuts

S.8. Reduce short-term memory load.

N.6. Recognition rather than recall. Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another.

N.7. Flexibility and efficiency of use. Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

N.8. Aesthetic and minimalist design. Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

U.Visibility (Simplicity): Keep the most commonly used options for a task visible (and the other options easily accessible).

U.Efficacy (Efficiency): Accommodate a user's continuous advancement in knowledge and skill. Do not impede efficient use by a skilled, experienced user.

U.Shortcuts (Efficiency): Allow experienced users to work more quickly by providing abbreviations, function keys, macros, or other accelerators, and allowing customization or tailoring of frequent actions.

U. Supportive automation (Workload Reduction): Make the user's work easier, simpler, faster, or more fun. Automate unwanted workload.

U. Reduce memory load (Workload Reduction): Keep displays brief and simple, consolidate and summarize data, and present new information with meaningful aids to interpretation. Do not require the user to remember information. Allow recognition rather than recall.

U. Free cognitive resources for high-level tasks (Workload Reduction): Eliminate mental calculations, estimations, comparisons, and unnecessary thinking. Reduce uncertainty.

G. Responsiveness

L.6. Every action should have a reaction.

L.12. The user should always know what is happening.

L.13. The user should control the system. The system shouldn't control the user. The user is the boss, and the system should show it.

S.3. Offer informative feedback.

S.4. Design dialogs to yield closure.

S.7. Support internal locus of control.

N.1. Visibility of system status. The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

U. Feedback (Communication): Provide appropriate, clear, and timely feedback to the user so that he sees the results of his actions and knows what is going on with the system.

U. Sequencing (Communication): Organize groups of actions with a beginning, middle, and end, so that users know where they are, when they are done, and have the satisfaction of accomplishment.

U. User control (Efficiency): Make users the initiators of actions rather than the responders to increase the users' sense that they are in charge of the system.

End of List

Looking at the above list you can immediately recognize, that

1. The four sets sometimes repeat and sometimes complement each other.
2. Different authors are more focused on different dimensions.
3. Attractiveness and Utility dimension almost completely went out of consideration.

A cause of the last observation is simple. Emotional Design is a relatively young area of research started in 2003 by Donald Norman's book with homonymous title [3]. Utility matter, on the contrary, is as old as the world and may be regarded as self-evident. However, it was rather recently found to be not as trivial as it seemed before (see, for example, current works on "User Experience ROI" by Susan Weinschenk et al., presented e.g. on www.humanfactors.com).

The same way we can use the dimensions for structuring usability patterns, methods and measurements.

VII. Measuring usability: Overview

Measuring usability was never considered to be an easy matter, but that's the primary purpose of defining a model. As Douglas W. Hubbard puts it in his book How to Measure Anything, "There are three reasons why people think that something can't be measured. Each of these three reasons is actually based on misconceptions about three different aspects of measurement: concept, object and method" [4].

Definitions of six dimensions declared earlier in this paper give us proper objects of measurement. And, for the concept, I accept Hubbard's definition of measurement as "A set of observations that reduce uncertainty where the result is expressed as a quantity". This concept actually means that any significant reduction in uncertainty is enough to make a measurement valuable, even if it doesn't eliminate the uncertainty completely. What remains is a method for acquiring and presenting usability measurements.

VIII. Measuring usability: Normalization

First of all, to make measurements on heterogeneous dimensions comparable, we need a common normalized scale. I suggest using a ten-point scale where measurements may take one of the following values: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, X. Here X denotes the highest possible value of a dimension and stands for Roman ten or eXcellent, whatever your prefer. This is the ultimate level of perfection in any dimension of usability, which is considered to be hardly achievable and thus will be rarely seen in actual measurement. The purpose of having X instead of 10 is to make a measurement representable as a single character. Then we can use the ten-point scale for defining the vector of usability in a form of [578436] where each number is a quality measurement of an artifact in a corresponding dimension of the CAUSER model. In the above example Comprehensibility=5, Attractiveness=7, Utility=8, Safety=4, Efficiency=3 and Responsiveness=6.

Once coordinates of an artifact in each dimension are defined, we can calculate the overall usability as a production of all six measures. An overall usability of [578436] thus would be 5*7*8*4*3*6=20160 comparing to 1 000 000 - the ultimate level of usability on the scale ([XXXXXX] vector). The point here is that we should multiply, not simply total or average them. Otherwise we would obscure the impact which low value in any single axis makes on overall usability. For example, if we'd total or average, usability measured as

[909999] would be higher than [777777]. While a common sense wisdom tells us the opposite. A zero or very low measurement in any single dimension immediately diminishes value of measurements in other dimensions. Regardless of other qualities, people would normally do their best to avoid using a system if it is virtually incomprehensible, absolutely useless, mortally dangerous, ludicrously inefficient, dead irresponsive or simply disgusting.

This calculation supports another two ideas. First is that focusing on improvements in the lowest measured dimensions adds more value to the overall usability. If, in the previous example, we improve Attractiveness by two points [598436], we will have 25920 for the total usability. While improving Efficiency by the same 2 points [578456] will give us 33600 as a result. Doesn't it look like a fair measurement? The second idea is that making substantial improvement to an apparently bad thing is relatively cheap. While, as we are approaching the "ideal" X measurement in every dimension, the effort for making every next step of improvement grows rather exponentially than linearly. If we measure ROI as usability improvement divided by effort, this kind of justification is becoming more obvious. For example if we make a transition from [888888] (=262144) to [999999] (=531441), we double usability, while transitioning from [222222] (=64) to [333333] (=729) increases it by an order of magnitude.

Such form of representing results of measuring is a useful step in reducing the level of uncertainty by narrowing its areas. E.g. if we don't have a method of measuring some dimension for a particular artifact, we can denote it with "?". Thus the [5??436] vector would mean that we cannot properly measure Attractiveness and Utility of the artifact, while we are perfectly certain about its other four dimensions.

IX. MEASURING USABILITY: DISAMBIGUATION

The next step in reducing uncertainty is disambiguation of the scales by defining their extreme points and, wherever applicable, intermediate points. These definitions can be used as reference points for expert judgment.

A. Comprehensibility

0 – a prolonged formal training and a vast experience is required to understand the system.

10 – a first-time user is able to recognize familiar elements and start using the system immediately.

Intermediate points of the Comprehensibility scale could be defined as:

2 – a user is able to understand the system by reading a documentation.

4 – a first-time user is able to understand the system by consulting with more experienced users.

6 – a user is able to understand the system using available hints and instant help.

8 – a user is able to understand the system by exploring its behavior.

B. Attractiveness

0 – every target user is considering the system to be disgusting.

10 – all target users absolutely love the system and happy to use it.

C. Utility

0 – the system is absolutely useless.

10 – the system allows a user to achieve a whole set of desired outcomes with all their variations.

D. Safety

0 – an occasional mistake in using the system can cause an unrecoverable disaster.

10 – using the system is absolutely safe for even first-time users.

E. Efficiency

0 – achieving a desired outcome is much easier without the system.

10 – the system makes achieving a desired outcome absolutely effortless.

F. Responsiveness

0 – a user is never sure what and when will come out of the system and is constantly wondering whether the system is doing something or simply hanging.

10 – the system gives immediate feedback on every user action and behaves just like if it was reading the user's mind.

X. MEASURING USABILITY: CONFIDENCE INTERVALS

Most often results of measuring usability will exist in a form of random samples coming from laboratory test, user feedback or expert judgments. To make these results usable, in analyzing them we should strive for being rather accurate than precise ([2], [4]).

Even though precise measurement of some usability attributes may seem to be theoretically feasible, in practice inaccuracy always comes from the subjective nature of usability as an object of measurement. Thus, if we want to be accurate in our awareness of that inaccuracy, whatever measurement we get, we should consider it as an approximate estimation rather than an exact calculation.

For example, if we could precisely enumerate desired outcomes of a system, we could measure Utility as a ratio of the number of outcomes which the system can potentially produce to the total number of user-desired outcomes. However, looking through a "microscope", we would soon discover that, what we were considering to be a single outcome X, actually has slight variations X1 and X2, which make a huge difference for a user.

When it comes to processing results of statistical sampling, a common temptation is to proceed with calculating a median and dispersion of the measured values. These two numbers tell us about a width of a confidence interval (CI) and its center position. It's also tempting to make as many observations as possible to increase precision of these calculations. However this is not what we need in a practical sense.

What we actually want is an ability to identify precise boundaries of the interval where our judgment is at least 90% certain. And we don't need hundreds of tests for achieving it, since with just five random samples we are already 93.8% confident that all other measurements will fall between the lowest value and the highest value of those five [4]. Obtaining more observations still may make sense if we are attempting to narrow down the interval or increase our level of confidence about the interval to 95% or 99%. The important point, though, is that we can very meaningfully proceed with just five measurements.

How we can practically apply the Hubbard's "rule of five"? Taking the largest and smallest values out of five measurements on each dimension gives us two vectors, corresponding to the worst and the best estimations of usability. In the earlier example usability of [578436]=20160 would more probably look like [445225]:[688657]. Here the worst estimate of usability is [445225]=1600, and the best estimate is [688657]=80640.

The above is an important discovery. A big gap between the best and the worst estimations of usability points to another problem – unreliable usability. We cannot effectively improve quality before stabilizing it. So we should focus on narrowing the confidence intervals in the most problematic dimensions. In the above example, by subtracting one vector from another we get CI=[688657]-[445225]=[243432]. After finding that the most problematic dimensions here are Attractiveness and Safety (with 4-points wide confidence intervals), we should further proceed with finding out why people give so different estimates of their level.

XI. CONCLUSIONS

The CAUSER model is suggesting a structured approach to understanding and practicing usability by decomposing it into six dimensions: Comprehensibility, Attractiveness, Utility, Safety, Efficiency and Responsiveness with a common normalized ten-point scale.

We can use this model to structure our formal and informal knowledge about software usability, as well as to choose and apply appropriate methods for measuring and improving usability in corresponding dimensions.

Independent measurements of specific usability attributes can be naturally composed together for getting a meaningful integral assesment of an overall usability. The results of these measurements can be expressed in a "vector" form as well as in a form of a single number.

We can effectively reduce our level of uncertainty about usability measurement using simplified statistics for identifying 90% confidence intervals in each of the dimensions. This, in turn, helps in increasing a return on our usability design efforts by focusing them on the most critical areas of improvement.

XII. EPILOGUE: NOT ONLY SOFTWARE

Similarly to software and technical systems, we can apply the CAUSER model for structuring and improving our experience with organizations and other people. We can identify companies and organizations as more or less usable in each of the six dimensions. And, when it comes to people, we would stick with those who are easy to deal with, attractively looking, clearly speaking, helpful, safe, and are immediately responding to our questions and needs. On the contrary we tend avoiding those people who are dangerous, irresponsive, bad looking, difficult to deal with, do not satisfy any our need or simply incomprehensible. And, when we love somebody, we tend to be forgiving to his or her minor deficiencies.

REFERENCES

[1] Alan Cooper, Robert Reimann, David Cronin, About Face 3: The Essentials of Interaction Design. Wiley, 2007.

[2] Charles J. Bodenstab, Information Breakthrough. Bookworld Services, 1997.

[3] Donald A. Norman, Emotional Design: Why We Love (or Hate) Everyday Things. Basic Books, 2005.

[4] Douglas W. Hubbard, How to Measure Anything: Finding the Value of "Intangibles" in Business. Wiley, 2007.

[5] Eric Schaffer, Beyond Usability: Designing for Persuasion, Emotion, and Trust. Whitepaper, Human Factors International, Inc., September 25, 2008.

[6] Hugh Beyer, Karen Holtzblatt, Contextual Design: Defining Customer-Centered Systems (Interactive Technologies). Morgan Kaufmann, 1997.

[7] Jeff Johnson, Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules. Morgan Kaufmann, 2010.

[8] JoAnn T. Hackos, Janice C. Redish, User and Task Analysis for Interface Design. John Wiley & Sons, Inc., 1998.

[9] Qianli Xu, Roger J. Jiao, Xi Yang, Martin Helander, Roger J. Jiao, Halimahtun M. Khalid, Damai Sciences Sdn Bhd, Anders Opperud, "An analytical Kano model for customer need analysis", Design Studies Vol 30 No. 1, 2008.