Value Enabling Interaction Mediates Between Design and Evaluation

Gilbert Cockton
School of Computing & Technology,
Sir Tom Cowie Campus, University of Sunderland,
St. Peter's Way, Sunderland SR6 0DD, UK.
Gilbert.Cockton@sunderland.ac.uk

ABSTRACT
The interplay between usability evaluation and user interface design is indirect and must be mediated by value enabling interaction. We do not evaluate systems in HCI, we evaluate interaction. We thus cannot evaluate designs, but only their consequences for the quality of interactions. In evaluating interaction, we anticipate or observe user difficulties. A design may or may not contain the potential causes of a user difficulty. Causes have to be inferred from user difficulties in context. There is thus no direct interplay in either direction, either from design to evaluation, or from evaluation to design. Instead, both are mediated by interaction, but even this mediation is not direct. We must reason from designs to interactions, and from interactions to design features as causal factors. However, these processes are inherently descriptive. The role of evaluation must go beyond description to judgement, since the literal meaning of "evaluation" is to *(bring) out value*, that is, to find it in one place and to express it somewhere else. In HCI, we find value in interaction, but we judge value in the world. Until we start by stating the intended value of digital products, HCI can not reach the end point of delivering computer systems that are *worth* using. The relationship between design and evaluation is thus mediated by user interactions that do (not) deliver intended value.

Categories and Subject Descriptors
ACM: H.1.2 – User/Machine Systems

General Terms
Design, Economics, Experimentation, Human Factors,

Keywords
Value-centred HCI, Design, Evaluation, Mediation.

1 INTRODUCTION
The workshop title "Improving the Interplay between Usability Evaluation and User Interface Design" implicitly, if not explicitly, assumes a direct relationship ("interplay") between design and evaluation. No such direct relationship exists. In reality, the relationship must be, not *interplay*, but a *chain of mediations* via user interaction and the intended value for a digital product.

NordiCHI 2004 Workshop on Improving the Interplay between Usability Evaluation and User Interface Design, October 24, 2004, Tampere, Finland.

2 VALUE AND E*VALU*ATION
The English word *evaluate* is a back-formation from the French *évaluation*, which in turn is formed from the French *évaluer*, that is *é+valuer*, which literally means to (bring) value out of (from the Latin prefix, *ex*, which here became *é*).

Value and e*valu*ation are this inextricably linked, and it is thus somewhat unnerving that this has hardly been mentioned in over three decades of HCI research and practice. The sense of "evaluation", like many English words, has broadened, so that the Freesearch on-line dictionary [9] defines it as:

> ***to judge or calculate the quality, importance, amount or value of something:***

The tendency within HCI has been to see evaluation as mostly a question of quality, sometimes of degree (amount) and of importance, but rarely of value. However, I will argue that it is possible to have quality with neither value nor importance, especially where quality is assessed with respect to generic standards and measures (amounts of errors etc.).

Interestingly, the Concise English Dictionary [12] has a weaker first sense for *evaluation* as "assess, appraise" and a second mathematical sense as calculation of some form. For many who use the word, "evaluation" has lost its clear connection with "valuation". I will argue that effective evaluation in HCI should be understood in terms of intended value for digital products and services. Value here is not necessarily commercial. It can be personal, spiritual, experiential, organizational, political or cultural. Value-centred HCI must thus be able to cope with a wide range of human values. The core skills here are the ability to express intended value, the ability to relate this via envisaged interaction to design decisions, and the ability to relate value to the planning and interpretation of system evaluation.

3 THE ARGUMENT FOR VALUE-CENTRED HCI
The argument is a historical one. We have exhausted objective and descriptive approaches to HCI. Over three and a half decades, we have moved through three foci for HCI: the system, the user and the context of use [4]. None of these can function adequately as the sole focus for HCI [3].

3.1 The System as Focus

Early HCI work focused on design guidelines. This tradition has continued, and many still act as if universal "one size fits all" solutions are possible for interactive systems. Such design rules are rules about system features. The assumption is that such features can be directly evaluated, but, from an HCI perspective, they cannot be. In HCI we evaluate interaction. It is difficult to imagine what evaluating a system could mean in human terms. The attributes of systems that can be directly evaluated concern internal, rather than external quality, that is, qualities such as performance efficiency, correctness, modifiability and maintainability [10].

A system or design can be described. Claims (often wild and unrealistic) can be made for systems or designs. However, in HCI, we can only evaluate *usage*. We look at the interactions between people and systems. While this may be obvious to evaluation experts, it does not stop people outside of HCI (and too many within) from acting as if designs can be evaluated and that quality can be encapsulated in good features.

We need to understand how such an illogical situation persists, i.e., a belief in quality *within* a digital artifact rather than the user experience. The origins of the belief may lie in the origins of computer science. These are more than harmless philosophical concerns: they lead to damaging technological utopianism and a fetishism of technology alienated from its human context.

A system-centred approach is a natural consequence of the mathematical Platonic mind-set in Computer Science. Many mathematicians believe in mathematical *discoveries* on the basis that there is a single fixed mathematical reality that is revealed through mathematical investigation. Mathematical objects, although wholly abstract and apparently constructions of human imagination, are held to *exist*, almost in the sense that physical objects exist, except that they cannot be directly perceived (i.e., they are not sensuous). These ideal forms have fixed inherent properties that are the essence of mathematical objects.

This Platonic view has severe consequences for HCI when transferred to computer systems, since mathematically inclined technologists are inclined to treat software as a mathematical object with fixed inherent properties. This manifests itself in HCI in the form of design principles, patterns and guidelines. While these can be contextualized, the overwhelming tendency is for design principles, patterns and guidelines to be stated as "one-size-fits-all" absolutes. The result is that human agency, individual differences and usage contexts are removed from the equation. This isolation, or estrangement, of humans from the properties or *qualities* of computer systems is a form of alienation, which has some of the key consequences outlined by Marx in the Paris Manuscripts [11]. Systems are described as *fetishes* with *totemic* qualities, just as commodities become fetishes by the alienation of human labour from its products. Marx's analysis is quoted and summarized as follows [8]:

> "A commodity appears at first sight an extremely obvious, trivial thing. But its analysis brings out that it is a very strange thing […]." Fetishism in anthropology refers to the primitive belief that godly powers can inhere in inanimate things (e.g., in totems). Marx borrows this …to make sense of what he terms "commodity fetishism" … the commodity remains simple as long as it is tied to its use-value. When a piece of wood is turned into a table through human labor, its use-value is clear and, as product, the table remains tied to its material use. However, as soon as the table "emerges as a commodity, it changes into a thing which "transcends sensuousness"… People … thus begin to treat commodities as if value inhered in the objects themselves, rather than in the amount of real labor expended to produce the object. What is … a social relation between people … instead assumes "the fantastic form of a relation between things".

We see very similar processes in operation with system-centred HCI. Once quality is seen to reside in systems, magical claims follow thick and fast. Within the history of HCI, we have been told that graphical user interfaces were inherently easy to use, that on-line agents will solve all our shopping dilemmas, that location-based services will bring us desparately sought information. In all cases, the new technologies will automatically deliver a technical utopia in all contexts for all users. The consequences of computer science thinking are explored further in my NordiCHI plenary [5].

To some extent, the first two questions for the workshop construct design products as things with intrinsic properties:

(1) Which products of interface design are useful as the basis for usability evaluations?

(2) How do the specific products from interface design influence the techniques that are relevant for the usability evaluation?

The answer to the first, given that we cannot directly evaluate systems, is "none". The answer to the second is that "they should not". We evaluate interaction, and what we thus require from design is the ability to contribute to the direct evaluation of interaction. There are two forms of design products that can do this. Firstly, some can be tested with users, such as paper mock-ups, wire frames or prototypes of varying fidelity. Secondly, some can be combined with contextual research and HCI knowledge to produce models or descriptions of potential interaction, which can then be evaluated (e.g., task models for GOMS or task descriptions for Cognitive Walkthrough).

What is key about design products is how well they let us create actual or imagined interactions. Actual interactions arise when evaluation participants interact with design products. Imagined interactions arise when we derive interaction sequences from design products. As long as we can create actual or imagined interactions, then design products are compatible with evaluation. The quality of evaluation depends in part on the quality of the created interactions, but the key to evaluation is understanding value, and this is wholly independent of design products. Value pre-exists and post-endures design and interaction. It should thus be possible to plan much of evaluation before any design product at all exists in any form.

In summary, system-centred HCI is illogical. Systems cannot be evaluated, only interaction can be. To support evaluation, design products must be able to either produce real interaction, or support the synthesis of predicted interactions. There is no direct interplay between design and evaluation. Both must be mediated by actual or imagined interaction. What we thus require are methods that situate design products within usage

interactions. Not surprisingly, this is how HCI evolved in the 1980s.

3.2 The User as Focus

System-centred HCI was succeeded by user-centred HCI. User testing and inspection methods were a key part of this progression. Usability evaluation came to focus on quality in use. Users' difficulties when interacting with a system would be observed and described. User-centred HCI moved from misguided attempts to evaluate software systems to evaluating the quality of interaction associated with a specific design. However, user-centred HCI doesn't really *evaluate* interaction, nor can it always link back its 'results' to design features.

Usability engineering approaches rarely *really evaluate* since they have no concept of intended product *value*. They thus cannot properly prioritize user difficulties. Generic severity scales (e.g., [14]) are not appropriate. Thus one may think that task failure is always the most severe form of "usability problem", but severity here actually depends on how critical the task is to delivering the intended value of a digital product. In some contexts, task success with residual errors (e.g., in the design of a safety critical product) is more severe than task failure. Non-existent designs are infinitely less harmful than dangerous ones.

Usability engineering tends to be context-independent. While user test scenarios may attempt to recreate real contexts of use, the results of user testing may not be reported back in a contextually sensitive manner. Error counts, time on task, success rates and subjective response can be treated as universally relevant measures. So in answer to the question:

(3) In which forms are the results of usability evaluations supplied back into interface design?

The answer is often "a useless one", i.e., the results of user testing take no account of what a product or service is trying to achieve. While in practice, experienced usability specialists do take business and other client goals into account when reporting, no existing method makes clear use of statements of intended value as inputs to planning and reporting. SUPEX [6] provides a 'filtering' hook to accommodate product goals, but no more. The result is that our publicly documented methods only aim to deliver quality in use. Achievement of product value is a matter of luck.

Without explicit inputs for intended value, usability engineering methods cannot be seen as *evaluation* methods. They assess and appraise in the weak sense of "evaluation", but they are not focused on establishing the impact of the user experience on the achieved value for a digital product or service. This impact occupies a continuum from destruction to donation. User experience may be so poor as to *destroy* all intended value. Conversely, it may be so surprisingly good that it *donates* unexpected value, i.e., both product sponsors and customers get more than they expect, which is the true mark of *gifted design*.

In between destruction and donation, user experience may *degrade* or *deliver* intended value. Where we cannot understand or fix catastrophic problems (because the technology simply can not work as hoped), then we must *deny* the possibility of a design ever delivering its intended value. Such reality checking is a common role for human factors experts, especially in response to naïve technological utopianism. However, denial is based on the absence of credible fixes and thus goes beyond the scope of evaluation. It is rather an issue for the *iteration* of designs.

There are thus '5 Ds' of HCI: deny, destroy, degrade, deliver and donate. The last four are a basis for *value impact analysis*. They do not feed through directly into design, and nor does the first, other than stopping all further design.

The purpose of evaluation is to assess value. It is not the role of evaluation to propose design changes. We need to be clear in distinguishing the description of interaction (the 'results' of user testing) from its evaluation (which assesses impact of user interaction on achieved product value). Evaluation results are thus not directly 'supplied back' into design. Instead, they isolate and identify the user difficulties (actual or imagined) that really matter. Design change recommendations need to be based on a credible causal analysis of user difficulties. This is *not* part of evaluation. The purpose of evaluation, once again, is to assess achieved value. *Explaining* why value is or is not achieved is a very different activity to *assessing* the achievement of value.

Thus in response to:

(4) Which usability evaluation results are needed in interface design?

The initial answer is that *value impact analysis* will identify user difficulties that destroy or degrade the achieved value of a digital product or service. However, progressing from this identification to the design changes that may deliver or donate value requires two distinct steps that are not part of evaluation activities. Instead, they are part of the *iteration* activities that move a design from one combination of value to a (hopefully) improved one. Dennis Wixon limits effective evaluation to two questions [1]: Do we understand the problem? Can we fix it? Reports of user difficulties are not sufficient for either. However, neither of these questions are part of the evaluation process, which should stop with identification of user difficulties that degrade product value.

User-centred HCI has provided little systematic support for iteration, which requires two distinct activities. The first is *causal exploration and analysis*, which may require more formal studies (even controlled experiments) to establish the causes of value degrading user difficulties. Evaluators need to work in collaboration with developers to properly structure causal analysis (often a developer will immediately understand why a difficulty has arisen, but an evaluator could take hours to reconstruct a causal chain).

The second iteration activity is *design change recommendation*, which requires extensive knowledge of interaction design and a full understanding of the goals for a product or service. An evaluator may not have all the knowledge and skills required to make credible design changes without the collaboration of software and project specialists.

We should thus separate evaluation from iteration. Evaluation should report in terms of value, and not in terms of generic error counts, stories of unhappy users and time on task — except

where these measures and information have a direct bearing on intended value.

Design iteration requires not only confidence in the results of usability evaluation, but also information that is directly relevant to making design decisions. Observations of user difficulties are only part of the analysis. Re-design requires a sound understanding of how design features combine with usage contexts to degrade the user experience.

In summary, current usability reports are not well focused on value, nor do (or should) evaluation methods be the main palce for causal analysis that can directly identify how users and design features interact to produce (un)acceptable interaction. Evaluation methods fail to provide what is needed, which is an evaluation in terms of 4 Ds of HCI (destruction, degradation, delivery and donation). Evaluation ends here. Iteration begins with the search for explanations of the impact of interaction on product value in terms of causal chains between user behaviour and system features. Iteration may end with the denial that intended value can be achieved with a target technology.

Dissatisfaction with 1980s user testing approaches [13], especially overreliance on generic measures such as error counts and time on task, led to the next major paradigm shift within HCI. However, while the move from system to user led to significant progress within HCI, the move from user to context did not address the main requirements for true evaluation: a focus on value.

3.3 Context of Use as Focus

The move from user- to context-centred HCI in the 1990s enabled more contextually sensitive and appropriate evaluation measures. These were (and are) more suitable inputs to value impact analysis.

The focus moved from the minutiae of quality in use to major issues of the fit between a design and an intended context of use. Contextually realistic evaluation increases confidence in the validity of reported user difficulties, but it does not move testing from appraisal/assessment to true evaluation.

Contextualised descriptions of user difficulties and interaction misfits are broader and more specific, and thus provide a conceptually richer space for explanation. As a result, context-centred HCI is better placed to understand what it is that intended users will actually value. Contextual research can be focused on understanding value in a way that psychological laboratory testing cannot. However, context-centred HCI has tended to focus on the 'fit' between a design and its intended context of use [3]. As with user difficulties, not all misfits have major consequences for the delivery of intended product value.

Context-centred HCI, as with user-centred HCI, lacks critical 'noise filters' that will focus evaluation on the delivery and enhancement of intended value. In electronics, a noise filter (such as Dolby™ tape noise reduction) removes noise from a channel, leaving mostly signal. We need something similar in HCI to isolate important problems from 'noisy' usability 'non-problems' and trivial inconsequential misfits.

In summary, contextual approaches have focused on (mis)fit, without necessarily addressing value or importance. Even so, they do identify potential loss of value that cannot be identified by traditional user testing. We must thus see evaluation as not only focussing on quality in use, but also on fit to context.

3.4 Answering Other Workshop Questions

From the above, my answers to the remaining questions could be very predictable:

(5) Do existing evaluation methods deliver the results that are needed in user interface design?

(6) How can usability evaluation be integrated more directly in user interface design?

(7) How can usability evaluation methods be applied in emerging techniques for user interface design?

My answers are:

(5) No, and it's not their role to. This is the role of iteration processes within development.

(6) Design and evaluation need to be integrated within a wider value-centred framework for HCI. Iteration is one key link between evaluation and design. The initial link is provided by *opportunity identification* processes. However, designs may need to incorporate support for evaluation.

(7) They cannot, evaluation and design need to be integrated within a wider value-centred framework for HCI.

This paper thus reframes the workshop problem. The relevant questions are not about direction relationships between design and evaluation, but instead about how design and evaluation relate to iteration and initial development in a value-centred framework,

4 A FRAMEWORK FOR VALUE-CENTRED HCI

The workshop focussed on the interplay between design and evaluation. In analysing the relationship between them, we need to bear the following in mind:

(1) Design and evaluation are complex processes that each require co-ordination of discrete activities

(2) We evaluate interaction, not design, so one of the interfaces between design and evaluation activities is the generation of actual or imagined interactions

(3) Evaluation should focus on the achievement of intended value, and thus statements of intended value are another interface between design and evaluation activities

(4) Evaluation does not (and should not) generate design recommendations, which are again the result of a process of co-ordinated *iteration* activities that begin with identification of destroyed and degraded value and end with design change recommendations

(5) Evaluation planning can commence once statements of intended value are available. Initial evaluation activities can be completed before any design activity commences. Only planning of precise evaluation procedures requires design products to establish the fine detail of evaluation.

(6) There are thus four broad processes in interactive systems development: design, evaluation, iteration, and *opportunity identification*, which must be completed before design and evaluation can begin. Similarly, iteration follows the completion of evaluation.

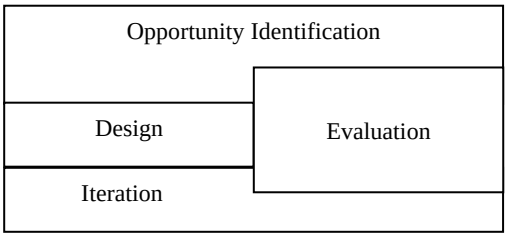

Figure 1. Main process structure for interactive systems development

Figure 1 shows the relationship between these development processes. Horizontal relationships indicate processes that can proceed in parallel. Vertical positions indicate logical dependencies, i.e., a process instance above must complete before the process below can start. Thus design and evaluation require statements of intended value as an output of opportunity identification to commence. Evaluation can start before design, and cannot complete before design (as this must pause to allow evaluation to take place). Similarly, evaluation must complete to allow iteration to commence. Once iteration is completed, development can recommence with new design and evaluation instances, or even with a revisitation of opportunity identification.

Figure 2 shows the internal structure of the four development processes within the context of a value-centred framework.Boxes represent products of development activities. Arrows represent activities which generate new development products from existing ones. Figure 2 is simplified. Arrows are labelled (e.g., E1), but not are all shown. For example, there should be arrows for activities that make use of *Design Change Recommendations* to update *Interaction Designs* and *Value Delivery Scenarios*, since causal analysis may have revealed poor decisions in any previous design activity. Similarly, causal analysis needs to be grounded in information on usage contexts, so a long back arrow is missing here.

Each arrow represents an activity performed by a development role. Thus statements of intended value are *derived* from representations of the context of use (activity O2), and are in turn inputs to both the *creation* of value delivery scenarios (activity D1) and a *transformation* into evaluation criteria (activity E1).

Activities in each main process are now briefly outlined. Examples are given for two hypothetical web-sites: one for van hire, and one for a university. Both are based on real development activities in which I have been engaged. The former involved commercial usability evaluation, and thus cannot be reported in any detail. The latter is ongoing and thus cannot be used as a detailed case study until further development iterations have completed.

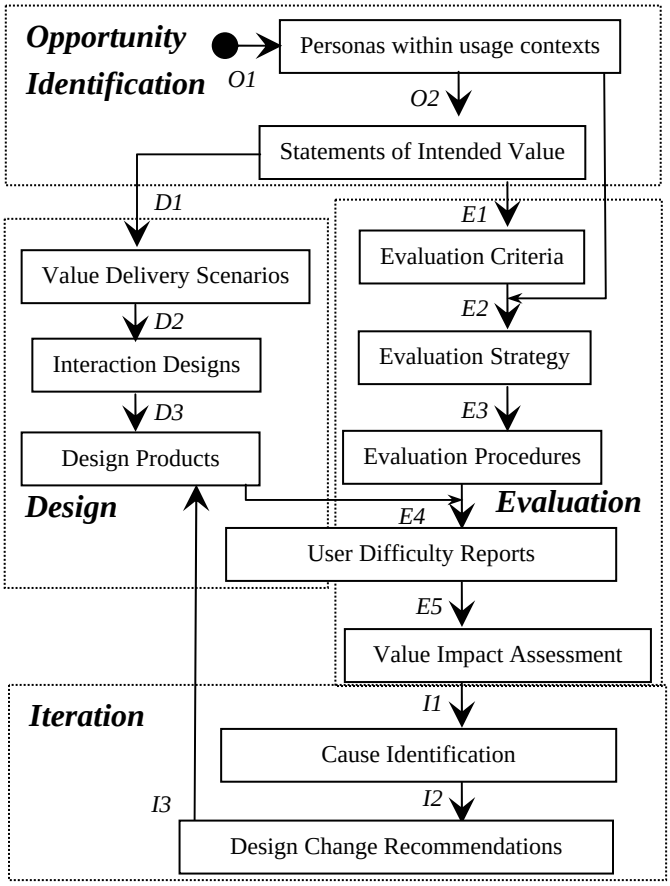

Figure 2. Activities and outputs within development processes for value-centred design

4.1 Opportunity Identification Activities

Opportunity identification is the process by which the intended value for a digital product or service is described and specified. It begins with *studies of usage contexts* (O1). These activities result in collections of models and descriptions of target usage contexts. The specific reference to personas [7] is deliberate. While most development products are general and should be able to accommodate a range of HCI methods, personas are highlighted as a method that are well suited to expressing the values of individuals and their organisations. Culture diagrams from Contextual Design [2] may also be appropriate forms for expressing value. Such development products are thus the main input to the second activity within opportunity generation: *intended value specification* (O2), which analyses contextual models and descriptions to identify opportunities for creating new value with a digital product and/or service(s). The result is a set of statements of intended value that should be delivered by a successful project.

For a university web-site, the key personas are university management, students, parents and career advisers. The primary value for the last three personas is the provision of appropriate, adequate and effective help with choice of course and university. For university management, a primary value from the web-site will be the achievement of high levels of student recruitment.

For a van hire web-site, the key personas are company managers, customers and depot staff, who respectively will derive value from: increased profits, improved brand equity, and recognised personal achievement; hiring an appropriate van for a suitable period at an economical cost as regards price and personal effort required to collect and return it; the smooth collection and return of vans by well-informed, well-prepared and satisfied customers.

The "intended value statements" for the two imaginary web-sites are very brief (e.g., "hiring an appropriate van for a suitable period at an economical cost as regards price and personal effort required to collect and return it"). The format and detail required for intended value statements is an open question in value-centred design. The sketches above are not sufficient, but at the same time, formats should be accessible for all stakeholders and the extent of detail is likely to be quite limited. Value can be stated succinctly.

4.2 Evaluation Activities

The evaluation process can begin before the design process, although both can run in parallel. The first activity, *value operationalisation* (E1) translates intended value statements into measurable evaluation criteria.

For a university web-site, example measurable criteria are the extent of engagement from site visitors (which pages get visited by who and what do they do as a result?) and the number of student enquiries, applications and enrolments that can be attributed to the university's web-site.

For a van hire web-site, example criteria are increased profits and improved brand equity attributable to the web-site. Although management desire recognised personal achievement and consequential career advancement, this is unlikely to be carried forward as an explicit evaluation criterion for a range of reasons that are easy to imagine. Further evaluation criteria are high levels of customer and depot staff satisfaction.

None of the example criteria above are usability requirements as would normally be understood. This is because quality in use and fit to context only matter in so far as they donate achieved value beyond what was sought, or when they destroy or degrade achievable value. Also, some of the example criteria cannot be addressed in existing usability testing approaches. Instead controls and measures must be placed *in the world, where value is achieved*, and not in the usability environment, which is transient and often artificial.

The second evaluation activity, *evaluation strategy formation* (E2), translates evaluation criteria into a strategy for monitoring and measuring the achievement of value. The main decisions here concern the choice of evaluation methods. User testing will be one part of this strategy, but evaluation has to extend to continuously monitoring the effectiveness of a system in real usage.

The third activity, *evaluation procedures design* (E3) selects measures and instruments for evaluation criteria that are appropriate for the evaluation methods selected as part of the evaluation strategy. Selected measures and instruments are associated with detailed procedures for each evaluation method.

The fourth activity, *evaluation implementation* (E4) applies evaluation methods to design products to produce reports of actual or predicted user difficulties. The fifth activity, *value impact analysis* (E5) assesses user difficulties in terms of their impact on achieved value. Only difficulties that destroy or degrade achieved value are carried forward for remediation during the iteration process. Note that value impact is not the same as severity. Most existing severity ratings are defined from a user/task perspective (e.g., [14]). However, value impact analysis has no pre-conceptions on whether task failure is always serious (it depends on the criticality of the task for delivering intended value), nor may moderate user disapproval be of limited concern (solid user approval may be vital to product success). What does and does not matter at this point is wholly dependent on earlier statements of intended value and their translation into evaluation criteria.

4.3 Design Activities

The first design activity, *value delivery scenario authoring* (D1) is similar to activity E1 (value operationalisation), as it restates statements of intended value in a form that can be used directly and effectively within subsequent activities in the design process. This activity refocuses existing HCI uses of scenarios to focus on the delivery of value in the world, rather than on quality in use and/or fit to context. It is guided by evaluation criteria that should be in place before scenario authoring is well advanced. Good scenarios here will be ones that tell plausible stories of how value results from envisaged designs.

For a university web-site, value delivery scenarios would explain how a proposed design would deliver appropriate, adequate and effective help with choice of course and university, and how this in turn would achieve high levels of student recruitment. Furthermore, once evaluation strategies are in place, value delivery scenarios should cover the details of how evaluation procedures will confirm the delivery or better of intended value. Thus the effectiveness of web content could be demonstrated via enquiry codes that link the web site into a university's marketing processes. Also, interactive content and downloads on the web-site could track prospective students from initial interest to making an on-line application. It would be possible to measure the attractiveness and effectiveness of web-site content. Usability evaluation would focus on quality in use, looking for interactions that degraded or destroyed intended value. The latter could indicate that the value delivery scenarios were misguided. Iteration would have to address this by changing scenarios as well as the design.

For a van hire web-site, value delivery scenarios would provide plausible stories on how proposed designs could increase profits and improve brand equity by letting customers hire an appropriate van for a suitable period at an economical cost as regards price and personal effort required to collect and return it. Other scenarios would tell stories of how site features ensure the smooth collection and return of vans by well-informed, well-prepared and satisfied customers.

With value delivery scenarios in place, the *interaction design* activity (D2) would create a set of interaction designs that would be used in the third *design implementation* activity (D3) to create design products. Design then halts until the evaluation and iteration processes have completed.

4.4 Iteration Activities

Iteration begins with *Causal Analysis* (I1), which seeks to identify the causes of user difficulties that destroy or degrade achieved value. The second activity *Design Change Recommendation* (I2) uses identified causes to generate design changes that should remove undesirable user difficulties.

Iteration activities require the involvement of all roles in development. Developers and designers need to support evaluators in the identification of causes of user difficulties. Evaluators' skills are of particular importance when further user testing or formal user studies are required to reliably identify the causes of user difficulties. The quality of identified causes is critical to recommending appropriate design changes. A change based on a faulty causal analysis is likely to not improve a design, and may even make it worse.

Designers, developers, marketing and product management need to be involved in design change recommendation. This is not a job that evaluators can carry out in isolation. Designers may have several untried options that could be tried for the next version of a design. Developers can identify the costs of various proposed changes. Marketing and product management can advise on the appropriateness of proposed changes in relation to the vision and goals for a product or service (i.e., they may be best placed to interpret intended value statements and relate these to proposed changes).

Change recommendations apply to all products of the design process. Scenarios, design rationales and details, as well as implementations, may need to be changed. The third iteration activity, *design change implementation* (I3), implements all necessary changes to any design product.

It may the case that no design recommendations can be made that can plausibly result in better delivery of intended value. In these situations, a project may have to be terminated. The possibility of achieving intended value is *denied.*

The outcomes of iteration are thus one of the following:

1. the addition of value to the outcomes of interacting with a digital product or service (an improvement on the donation of value, moving from delivery to donation of value, moving from destruction/degradation of value to degradation or delivery)
2. the termination of a project (the denial that intended value can be achieved through an apparently promising technology)

The 5 Ds of HCI can thus be used to assess not only the impact of user interaction on achieved value, but also the outcome of iteration and its associated design change recommendations.

5 CONCLUSIONS

There is no direct relationship between design and evaluation, which are complex, multi-activity processes that are mediated, initially by a process of opportunity identification, and lastly by iteration. Design and evaluation can proceed in parallel, but design will benefit from a timely consideration of evaluation criteria and evaluation strategies, especially when the latter embed evaluation instruments in the product.

The view that designs can be directly evaluated is the result of a dominating misconception in Computer Science, i.e., the view that objects have fixed attributes and inherent qualities that can be asserted on the basis of feature descriptions. The attempt to evaluate designs directly is a form of alienation that strips interactive systems from their usage contexts, attributing quality in use to artefacts, rather than to the interaction of real humans with their own technologies. Interaction in turn must not be evaluated as a thing-in-itself, as is the case with quality in use approaches. Nor must fit to context be seen as the end point of successful design. Instead, the aim of design is to create new forms of value, and it is the achievement of value in the world that we should be evaluating.

We therefore need to develop value-centred frameworks for interactive systems development. These require three novel development products, with associated activities:

- Statements of intended value
- Value delivery scenarios
- Value impact assessment

The move from existing development methodologies to a value-centred one is thus dependent on our ability to:

- devise formats for intended value statements
- author effective value delivery scenarios
- assess the impact of actual and predicted user difficulties and contextual misfit on achieved value

Value-centred development creates further challenges in separating evaluation from iteration. This highlights the limitations of existing usability engineering approaches to causal analysis and design change recommendation. These tend to get buried in the corners of existing evaluation methods, but once they are isolated and scrutinised, there is little of substance to them. By restricting evaluation to the assessment of achieved value, separate iteration activities are required to bring all development resources to bear on well grounded and broadly based design change recommendations. A clean break is needed from past muddling through, and identifying iteration as a distinct process in its own right allows this. A new research area is needed to establish effective and credible iteration methods.

The belief that we can improve some direct interplay between design and evaluation is a logical consequence of both confusing the processes of evaluation and iteration, and also of seeing evaluation as the direct assessment of systems rather than an analysis of the consequences of adverse interactions for the achieved value of a digital product or service. Once we realise that we must separate iteration from evaluation, and that we must evaluate, not systems, but interactions, and evaluate on the basis of achieved value, then we are clearly directed to value-centred development frameworks that are grounded on value enabling interactions rather than on the creation of inherently and intrinsically usable artefacts.

Value-centred HCI is at a very early stage. Much work needs to be done to move it from a set of arguments and potential approaches to a set of proven development approaches. However, it already has value (!) as a conceptual framework that reframes and clarifies several key issues in HCI.

6 REFERENCES

[1] Barnum, C., Bevan, N., Cockton, G., Nielsen, J., Spool, J., and Wixon, D., "The "Magic Number 5": Is It Enough for Web Testing?", in *CHI 2003 Extended Abstracts*, eds. G. Cockton *et al.*, 698-699. ACM Press: New York, 2003.

[2] Beyer, H. and Holtzblatt, K., *Contextual Design*, Morgan Kaufman, 1998.

[3] Cockton, G., "From Quality in Use to Value in the World", in *CHI 2004 Extended Abstracts*, ACM Press: New York, ISBN: 1-5811 3-703-6, 1287-90, 2004.

[4] Cockton, G., "Three and a Half Decades of HCI: Three Bricks Walls and Half a Ladder," in *Proceedings of HCI 2004*, Volume 2, eds. A. Dearden and Leon Watts, Research Press International, 17-20, ISBN:1-897851-13-8, 2004.

[5] Cockton, G., "Value-Centred HCI", Proceedings of the Third Nordic Conference on Human-Computer Interaction, ed. A. Hyrskykari, 149-160, ISBN 1-58113-857-1, 2004.

[6] Cockton, G. and Lavery, D. "A Framework for Usability Problem Extraction", in *INTERACT 99 Proceedings*, eds. A. Sasse and C. Johnson, 347-355, 1999.

[7] Cooper, A. and Reimann, R.M, *About Face 2.0: The Essentials of Interaction Design*, Wiley, 2003.

[8] Felluga, D., "Modules on Marx: On Fetishism." Introductory Guide to Critical Theory. http://www.purdue.edu/guidetotheory/marxism/modules/marxfetishism.html, Purdue University, last update: 28/11/03, last visited 8/10/04.

[9] Freesearch dictionary, http://www.freesearch.co.uk/dictionary/evaluate, last visited 8/10/04.

[10] Gram, C. and Cockton, G. (co-editors of monograph). Design Principles for Interactive Systems, Chapman and Hall, 248 pages, ISBN 0 412 72470 7, 1996.

[11] Marx, K., *Economic & Philosophical Manuscripts of 1844*, Progress Publishers, Moscow 1959.

[12] D. Thompson (ed.), *The Concise Oxford English Dictionary*, 9th edition, 1999.

[13] Whiteside, J., Bennett, J., & Holtzblatt, K., "Usability engineering: Our experience and evolution," in *Handbook of Human-Computer Interaction*, 1st edition, ed. M. Helander, North-Holland, 791-817, 1988.

[14] Wilson, C., "Reader's Questions: Severity Scale for Classifying Usability Problems," *Usability Interface*, 5(4), April 1999, available at http://www.stcsig.org/usability/newsletter/9904-severity-scale.html