2005

Increasing human-centred software development maturity in software development organizations

Ghassan Al Qaimari

University of Wollongong in Dubai, ghassanqaimari@uowdubai.ac.ae

Publication Details

Al Qaimari, G. (2005). Increasing human-centred software development maturity in software development organizations. In K. Soliman (Eds.), Internet and Information Technology in Modern Organizations: Challenges and Answers (pp. 73-80). Cairo: IBIMA.
Increasing human-centred software development maturity in software development organizations

Abstract
Interactive development requires the collaborative efforts of traditional software engineering and human-centred design (HCD). Organisations that develop interactive products must establish a multidisciplinary culture and integrate human-centred design into the software development process. Appropriately, selected metrics coupled with an effective measurement program assess an organisation's competency level in defining, practising and integrating human-centred activities into traditional software development. The capture and dissemination of experience affords benefits of reuse to be conferred organisation-wide. Quantitative assessment of human-centred process performance, as part of routine quality assurance activities, raises an organisation's human-centred maturity in the Usability Maturity Model capability scale.

Disciplines
Business | Social and Behavioral Sciences

Publication Details
Al Qaimari, G. (2005). Increasing human-centred software development maturity in software development organizations. In K. Soliman (Eds.), Internet and Information Technology in Modern Organizations: Challenges and Answers (pp. 73-80). Cairo: IBIMA.

This conference paper is available at Research Online: http://ro.uow.edu.au/commpapers/2912
Increasing Human-Centred Software Development Maturity in Software Development Organizations

Ghassan AL-Qaimari,
College of Information Technology, University of Wollongong in Dubai, P. O. Box 20183, Dubai, UAE
GhassanQaimari@uowdubai.ac.ae

Abstract
Interactive development requires the collaborative efforts of traditional software engineering and human-centred design (HCD). Organisations that develop interactive products must establish a multidisciplinary culture and integrate human-centred design into the software development process. Appropriately selected metrics coupled with an effective measurement program assess an organisation's competency level in defining, practising and integrating human-centred activities into traditional software development. The capture and dissemination of experience affords benefits of reuse to be conferred organisation-wide. Quantitative assessment of human-centred process performance, as part of routine quality assurance activities, raises an organisation's human-centred maturity in the Usability Maturity Model capability scale.

Keywords: Human-Centered Design, Usability Maturity, Quantitative Assessment, Knowledge Reuse.

1. Introduction
The inadequacies of existing software development processes in addressing behavioural aspects of interactive development have necessitated a human-centred design (HCD) approach. HCD is perceived as knowledge intensive because there are too many constraints associated with the practice of its activities. Generally in an organization, knowledge necessary to execute the HCD activities is often missing or not readily available. With no knowledge to serve as a baseline, the organisational effort is estimated as excessive. In addition, shortages in work resources, time, and cost are also used to justify the exclusion of human-centred activities or part of software development. Therefore organisations rarely practice HCD in its entirety, commonly a sub set of activities are practiced in the later phases of development, i.e., usability testing during the testing phase [2, 3].

The insights into organisational practices concerning human-centred activities expose the insufficiency of process support for HCD [2]. HCD activities are recognised as distinct activities that augment the numerous software development activities. Yet this shallow exploration of human-centred activities fails to reveal that some HCD activities follow, parallel, or overlap traditional software development activities. Therefore the definition and establishment of HCD within the organisation and its integration with traditional software development would in all likelihood increase and institutionalise the practice of HCD activities.

In this paper we present a process-based perspective designed to support human-centred software development by the Reuse of Experience and Metrics Assessment (REMA). REMA, which was first introduced in [14], is a heuristic environment that routinely seeks to support the quality assessment of an organisation's human-centred process. The principle behind REMA is to increase organisational understanding of HCD activities in an effort to circumvent human-centred process and product failure by logically reasoning from knowledge and metrics data. The collection and analysis of human-centred process performance cooperatively institutionalises a formal and consistent approach towards the desegregation of human-centred design and traditional software development.

2. Background
Usability
Formally, the International Standards Organization (ISO) defines usability as "...the effectiveness, efficiency and satisfaction with which the specified users can achieve specified goals in particular environments" (ISO DIS 9241-11). The effectiveness and efficiency of usability attributes refer to the extent to which the goal or task is achieved and the amount of effort required to accomplish the goal respectively [11]. Satisfaction is the subjective comfort level of users when interacting with the product including the aesthetics and acceptability in achieving user goals. To consider user issues and provide a usable and useful product an organisation ought to adopt a Human-Centered Design (HCD) approach to software development in collaboration with archaic Software Development Life Cycles

* ISO DIS 9241-11: Guidance on Usability
The Star Life Cycle (SDLC) is a user-oriented, iterative, non-ordered, phase-entered into. The evaluation phase is considered essential final arbiters of decisions [11].

Context and environment are all derived from the user’s interdependent process model, created to support the development process. Ideally the user is an active participant in all stages of development and with all product deliverable’s. A product’s goals, objectives, performance and preferences are the final arbiters of decisions [11].

In 1998 Barry Boehm [4] proposed a SDLC that introduced iteration, prototyping and risk analysis as new process characteristics to software development. The evolutionary nature of the Spiral Model meant that software development flow began in the centre and progressed outward towards a complete system. The design space was transformed into a myriad of alternatives that were not limited to previous decisions. Although an improvement to the limitations incurred by the Waterfall Model, the Spiral Model remains a SDLC that is primarily concerned with system functionality (constructional domain).

A Human-Centered Design (HCD) approach focuses on the behavioral aspects of the user when interacting with the system (behavioral domain). The essential principle of HCD places the user at the centre of the software development process. Ideally the user is an active participant in all phases of development and with all product deliverable’s. A product’s goals, objectives, context and environment are all derived from the user’s viewpoint, the user’s performance and preferences are the final arbiters of decisions [11].

The Star Life Cycle [10], an example of a behavioral SDLC, is a user-oriented, iterative, non-ordered, phase-interdependent process model, created to support the HCD principles of user-focus, iteration and prototyping [12]. Not imposing a specific entry or exit phase, software development may commence at any phase but before progressing to another phase the evaluation phase must be entered into. The evaluation phase is considered essential in all stages of development. Within the evaluation phase, user criticisms and suggestions are considered and the next phase is selected based upon user feedback. This is in contrast to pipeline architecture such as the Waterfall Model. The non-ordered phase structure of the Star Life Cycle enables a phase to undergo any number of iterations before progressing to another phase.

Behavioral SDLC models transcend classic SDLC models by introducing the user into the development process and by specifying how the functionality should be implemented in order to match user requirements [10]. This specification creates a bi-directional relationship between the product image and the user’s mental model of the product. The granularity of the bi-directional relationship is dependent upon the degree of human-centeredness in an organization. At present the vast majority of organizations are clustered at the ad-hoc end of the application spectrum of HCD activities. HCD activities are not considered to be an organization wide collaboration.

Knowledge Reuse and Knowledge Management

In many organisations, human-centred design is an ignored process as there are no accessible specifications concerning human-centred activities, methods or tools. Although there is a substantial body of publicly available knowledge, much of this knowledge is only known to specialists in the field. Therefore the publicly available knowledge is unable to be transferred into organisational ownership inhibiting the fostering of organisational learning of existing human-centred best practices. Accordingly there is a realisation on the importance of sharing knowledge. Knowledge of both experience and performance is effective at supporting the process, the project and the practitioner. Because it takes many years to develop experience in a knowledge intensive area such as human-centred design, it is often opportunistic and evolutionary to reuse existing knowledge as intellectual leverage instead of rediscovering the knowledge.

At present, the expression, organisation and distribution of human-centred organisational knowledge is a time-consuming and difficult task for an individual, a group or an organisation. Experiences developed in an organisation are typically not published to the entire organisation, rather experiences are bound to their author(s), inhibiting the dissimilation and accessibility of the available body of knowledge, experiences or the ability to capture experiences evaporates simultaneously to the departure of their author(s) [3]. As such, any HCD knowledge is perhaps an organisations most valuable resource.

Knowledge management (KM) encompasses the processes of identifying, acquiring, storing, organizing and dissimilating knowledge to provide a competitive
advantage within the organization. KM is aimed at creating an environment where knowledge may be shared organization-wide. Where individuals or teams are able to exploit lessons learned when they need to complete a task or make a decision [13]. Using existing knowledge (HCD.5.4) is an activity of process five in the Usability Maturity Model (UMM) [7] – (see next section).

The success of KM requires an organization wide commitment to a learning culture. It involves attention be brought to process methodology and individual issues relative to the business objectives of an organization. The perception, that human-centered activities are not practicable without explicit human-centered experience, is the motivation behind introducing KM. Common complaints concerning the inclusion or practice of human-centered activities include: excessive budget costs, additional training, schedule delay and requisite usability experts. At the individual level there are objections and resentment to process and mindset change. These are challenges and obstacles that need to be addressed when changing organizational culture to incorporate HCD. Organizations need to establish opportunities that demonstrate the value of KM as an active long-term investment in process, people and product quality.

The Usability Maturity Model (UMM)

The encouragement of human-centred activities into conventional software development requires an understanding of where the organisation is with regards to HCD activities. The Usability Maturity Model (UMM) [7], a stand alone process assessment model compliant with ISO standards 15504\(^1\) and 13407\(^2\), assesses an organisation's competency level in defining, practising and integrating human-centred activities. Ill-defined HCD processes results in less than successful practice of human-centred activities. Improvised practice of HCD activities results in sub optimal product quality. The assurance of product quality largely depends on the maturity of definition, practice and integration of human-centred activities. The UMM provides a basis for process planning, measure the capability of the process in use, or assist in the improvement of human-centred performance. The UMM does not specify any HCD process, rather it prescribes the characteristics that a human-centred process must exhibit to qualify as a process of some maturity.

The UMM is composed of seven processes\(^8\) each process may be categorised as either an organisational or developmental process. Organisational processes promote HCD awareness throughout the organisation, establishing a cohesive relationship between management and practitioners. Development processes describe the activities that should be performed in order to represent and include the users of a system during the development process [7].

An organisation may make use of the UMM to distinguish between immature and mature HCD processes. Immature processes are defined as ad-hoc, ill-structured processes where projects are executed without strategy, and the outcome is largely dependent upon the capability of the development team [6]. Conversely in an organisation with mature processes, a project is executed by following various existing organisational processes. Therefore, the outcome of the project is less dependent on team capability and more controlled by processes [6].

3. Organizational Tasks to Establish Human-Centered Process

There is an infancy of organisational knowledge concerning human-centred design. The majority of human-centred processes do not accurately reflect the attributes and activities described within behavioural software development life cycles. The tailoring of life cycle to process scope is more often than not unavoidably erroneous due to the shift in mind set, knowledge and experience from the constructional to the behavioural domain. Consequently REMA defines the following organisational tasks that require an organisation to establish succinct goals that mirror the organisational needs beseeching a human-centred approach to interactive software development.

1. **Describe the foreseeable role of human-centred design in the organisation.** Translate organisational obstacles into goals and characterise the areas of the organisation under focus. List the motivations behind the institution of HCD activities into the existing software development processes. Establish relationships between the organisational obstacles and the benefits expected from the inclusion of human-centred design.

2. **Document the existing or desirable HCD process.** Identify phases and define phase constituents. Select necessary activities (base practices) from

---

\(^1\)ISO TR 15504 Software process assessment.
\(^2\)ISO/IEC 13407 (1997) Human-centered design processes for interactive systems.

\(^8\) Throughout the remaining paper, processes, base and management practices shall be cited where applicable in context.
4. Reus-Enabled Software Development

An increase in human-centred process maturity is dependent upon knowledgeable decisions that are based on empirical evidence of success and failures. There is a need to know the reasoning and rationale behind these decisions, yet documenting the human-centred process is a time-consuming process in itself, therefore it is optimal to automate the process. Unfortunately, human-centred organisational experience is not routinely captured or disseminated. Organisations do not reuse existing knowledge as intellectual leverage to avoid common pitfalls or replicate successful endeavours. Rather the reinvention of the wheel in repeating old mistakes is the norm. REMA aims to capitalise on the reuse of organisational human-centred experience, to eliminate redundant efforts and streamline the desegregation of the behavioural and constructional software development processes.

Traditionally reuse initiatives have limited the experience type to concrete objects such as: source code, in contrast Basili [9], states that all experience generated from the development process including products, processes and other knowledge may be reused. According to Preece [4], it is practical to record crucial decisions which were difficult to make or were arguable. Decisions that compromise or hinder quality may be avoided. Conceptualisations...
behind decisions, alternatives and trade-offs establish a history. The history of project experience provides continuous accountability for decisions throughout development. Information may be retrieved to support management during process activities such as: risk analysis; project scheduling or resource allocation. Design wise, creativity is widened, alternate designs are explored and other ideas are generated. Financially, there is a reduction in development costs and time to market.

Basili and Rombach [9] proposed the reuse-oriented software development model as a means to encourage organisational exploration, feedback and sharing of software development knowledge. Two process models were identified, the archaic development model and the reuse model. In the development model the required object \( x \) is developed from scratch. In the reuse model, given a specification \( X \) for an object \( x \), an existing experience \( x_k \) is retrieved from an experience base and considered for reuse in favour of recreation of \( x \). The process of identifying and selecting a reusable candidate involves finding a set of candidates \( x_1, \ldots, x_n \) with the potential of satisfying reuse requirements \( X \). These candidates are then evaluated to derive at \( x_k \). Evaluation determines how well the available reuse candidates are qualified to meet the requirements \( X \); the degree of appropriateness [9]. If reuse of \( x_k \) is not appropriate, either the reusable candidate may be modified or several candidates may be merged into \( J \) and reused. Should the reuse process prove not to be successful, the required object \( x \) is developed from scratch, or a new attempt at the reuse process is commenced. A new attempt may require modifications to the reuse requirements \( X \).

A process that is not reuse-enabled or considers reuse to be of low priority may not endorse the decision to reuse existing experience. To ensure that the proposed experience and the reuse candidates are required, especially if there is a discrepancy amongst team members regarding their value and timing, a rating scheme of similar experience (if historical information is available) should be consulted and thereafter influence acceptance or rejection of the reuse candidates. The rating scheme exists as a means to judge if and when to reuse candidates. Given the possibility of reuse, the pre-use ratings give an indication of the perceived appropriateness of the reuse candidates \( x_1, \ldots, x_n \). Pre rating an experience should require the participation of at least one to two participants across actor categories involved in the software development. These may include a usability specialist; developers; a project manager and a team leader. Each actor is considered a stakeholder in the development of the product and may represent a different view regarding the benefit, cost and timing of the proposed candidates.

Evaluation may also be influenced by ratings given by persons involved with or affected by reuse. Practitioners may decide the usefulness of a reuse candidate with attitudinal bias to human-centred design. Alternatively usefulness may be decided upon by constraining factors on the process such as time and effort. Evaluation of reuse candidates cannot be done in isolation to the development team and the enclosing process. These variables play a significant role in the successful implementation of reuse initiatives.

Example

Take the example of an intelligent interface that goes beyond the direct manipulation style of interaction. In such systems "predicability" is an important usability attribute which may lead to certain usability requirements that will have ramifications on the design, rationale and the development process. At a certain stage in the development process, the usability engineer may want to recruit a sample of the targeted users to conduct an exploratory evaluation. Conversely the developer may not see the benefit of such activity due to a coming deadline. The project manager on the other hand may not be enthusiastic to approve the activity because of the costs involved, yet as a compromise may suggest a cognitive walkthrough analysis instead of an exploratory evaluation. At this stage it would be very helpful if the development team was able to draw a historical case which had similar usability attributes to the current project from the experience base.

Table 1: Historical Case

<table>
<thead>
<tr>
<th>Proposed Activity:</th>
<th>Exploratory Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate Activity:</td>
<td>Cognitive Walkthrough Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actor</th>
<th>Pre Rating</th>
<th>Post Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>9 / 10</td>
<td>9 / 10</td>
</tr>
<tr>
<td>Engineer</td>
<td>4 / 10</td>
<td>8 / 10</td>
</tr>
<tr>
<td>Developer</td>
<td>6 / 10</td>
<td>9 / 10</td>
</tr>
<tr>
<td>Project Manager</td>
<td>6 / 10</td>
<td>9 / 10</td>
</tr>
</tbody>
</table>

Assume a historical case with the ratings similar to Table 1. The ratings clearly support the argument of the usability engineer and should encourage the project manager to not only agree with the need to conduct a cognitive walkthrough analysis but also to recruit a sample of the targeted users and conduct an
exploratory evaluation. Historical information such as provided in Table 1 may be very valuable to a project if the context in the historical case is similar to the needing project and if both developments share many of the usability requirements.

As illustrated in Figure 1, evaluating a reuse candidate should not rely entirely on similarity comparisons of classification schemes or on descriptions and their correlation to the development. The usability attributes for a new interactive project should directly influence the selection of usability activities during its development process and the nomination of reuse candidates. Identifying the usability attributes of a new interactive system is crucial to its success. Based on the outcome of activities such as task analysis, user modelling and domain analysis, an experienced usability specialist should be able to articulate these usability attributes, and translate them into a list of usability requirements. Usability requirements directly influence the design, implementation and evaluation of the system. As a result the development team should be able to tentatively plan the sort of HCD activities required at every stage of the development process and nominate reuse candidates accordingly.

The experience base may propose experiences and the actors endorse or reject these experiences. Yet reuse of experiences should not be limited to the experience base, a qualified practitioner in usability may suggest an alternative activity or approach to the problem and may rely on the experience base to confirm his decisions or not (if there is a similar historical case for support). An organisation with a high commitment to usability should explicitly articulate the task, user and the domain, using the experience base for support. If in the case an organisation does not have access to a qualified practitioner of usability, the organisation may solely rely on the experience base. It is therefore essential to resist the urge to institutionalise reuse initiatives before the people and the processes have reached the proper maturity.

5. Quantitative Assessment of Process Performance

Improvements to human-centred organisational process maturity may be unambiguously determined through measurement and assessment: proper estimation; comparisons and predictions. Improper estimation is one of the causes of human-centred process failure. Ideally estimation, the base activity for effective project planning, should be performed early in the life cycle of the project as it affects other project characteristics. As project characteristics change, the associated estimations should change alongside. If not done, the estimations shall differ from the eventual actual values when estimations should reflect future expectations and should be derived from current project characteristics. Without accurate estimations, actual values cannot be compared for discrepancies, as inaccuracy does not permit a baseline for quantifiable comparison. As a consequence, modifications or improvements to the process in use, or to the quality of the estimation process are not advantageous. The comparison of estimated values to the actual values imparts the accuracy of the estimation process. If a large deviation exists then an investigation into the causes shall be instigated and a causal relationship between estimation efforts and actuality shall be established [5]. Collecting the specified measurement data from the implementation of the defined process (MP4.1.3)[8] is a key requirement in level four of the UMM.

The collection of metrics concerning process practice equips the organisation with knowledge or metrics data concerning its performance in human-centred design. On the basis of the quantitative understanding, the organisation may build a database of knowledge obtained from successful and failed practices. The knowledge of successes and failures and their causes
forms a baseline for future estimations and quantifiable comparisons of human-centred process practices. The baselines eliminate blind conjecture that is prevalent in human-centred design by establishing a foundation of reliance, in effect, supporting HCD process and product quality.

Quantitative assessment of human-centred process performance shall raise the process capability of an organisation in performing human-centred design. Moreover level four of the UMM stipulates process measurement in order for an organisation to arrive at a predictable process. Based on the metrics data obtained in level four, the practiced process may be modified or improved if need be. Level four and five of the Usability Maturity Model collectively instil a cyclical process of metrics measurement and assessment of HCD process practice. Quantitative assessment of human-centred process performance directly supports the HCD process, indirectly supporting the product.

6. Conclusion

The assurance of product quality depends largely on the quality of implementation and integration of the human-centred process within the organisation [8]. Interactive developments require the collaborative efforts of traditional software engineering and human-centred design in order to maximise the benefits gained from both human-centred design and traditional software engineering [1].

Ideally, human-centeredness is to be practiced throughout the entire software development process. Realistically, the lack of integration between HCD and traditional software engineering compromises the effective use of HCD in interactive developments [1]. Currently, traditional software engineering is dominative to human-centred design with minimal to no attention on user issues. Management do not establish, and practitioners do not maintain focus on user issues. Organisations that develop interactive products must establish a multi-disciplinary culture, where the organisation at-large, not simply individual advocates of human-centred design may see the added value of HCD in conjunction with traditional software engineering.

An established process for improving the processes of an organisation is to base enhancement of a process on experience gained from successful and failed developments [6]. Identifying successes and failures and their causes is reliant on quantitative data about process performance. Quantitative data regarding process performance considerably enhances process capability, by providing visibility into process omissions or activity relevancy across the breadth of projects. Based on the commonalities between process failures, the standard process definition may be subject to inspection. To identify and approve changes to the standard process definition on the basis of quantitative understanding of the process (MP5.1.1) [7] is a key requirement of level five in the UMM. MP5.1.1 implies that an organisation may define and trial more than one human-centred standard process to determine which is more appropriate. The assessment of which shall provide feedback into the standard process from experience of using the defined process (MP3.1.4) [7]. Therefore, over time the quantitative data shall assist in convincing the organisation of the returns on the investments made by the organisation in adopting a human-centred process and in iteratively improving process capability to achieve organisational goals.

7. Acknowledgement

This paper is based on earlier research work, conducted at RMIT University by Luisa Agi, supervised by the author of this paper, and reported in [14].

8. References


