FORMATION OF INDIVIDUAL HEURISTICS FOR SOFTWARE MAINTENANCE PROJECTS IN SMALL ORGANIZATIONS

Raza Hasan, Atiya Afsana
Towson University, Department of Computer and Information Sciences, Towson, Maryland
rhasan@towson.edu, aafsana@towson.edu

ABSTRACT

Due to lack of resources, small organizations rely heavily on individuals to achieve success in software maintenance projects. In this paper qualitative research approaches (Case Study and Grounded Theory) are used to show that these individuals utilize certain heuristics to carry out software maintenance projects. Data review and analysis show that these heuristics are developed over a formation process that begins with an individual looking at a problem or an opportunity, processing certain methods, and generating results. If the results are good then the same rules are repeated and with repetition they become individual heuristics. Based on empirical data, an individual heuristic formation model is presented that depicts this process.

Keywords: Small Organizations, Software Maintenance, Heuristics, Information Systems, Grounded Theory Method, Case Study Method

INTRODUCTION

Small organizations have been driving the economic growth in many countries of the world (Software Industry Statistics 1991-2005). Moreover, Internet has enabled small organizations to grow global and reach multi-million dollar turnovers (Poon, 2000). Software maintenance is a critical function for small organizations as 60-80 percent of organizational resources are allocated to software maintenance (Takang and Grubb, 1996). However, despite the significance of software maintenance in small organizations, little research has been carried out to investigate the software maintenance processes or challenges found in small organizations (Brodman and Johnson, 1994). Some of the specific challenges are lack of resources - personnel, time, skills and funds (Swanson and Dans, 2000; Benestad, Anda and Arisholm, 2009; Pigoski 1997). Such shortage of resources results in insufficient processes, methodologies, guidelines, tools and documentation needed for Software Maintenance. These deficiencies lead to major problems: Software Maintenance becomes difficult, complex, expensive, inefficient and unmanageable (Antquetil, De Oliveira, De Sousa and Batista 2007; Ko, Coblenz and Htet 2006). Again, such challenges and problems of small organizations are not given attention and few solutions have been seen; instead a majority of the software maintenance guidelines are geared toward large organizations. (April et al, 2005).

In small organizations, software maintenance processes are carried out by neither set procedures nor totally chaotic mental influxes; rather they are steered by certain heuristics that are developed by individuals over time (Anonymous, 2011). Since we did not find any discussion in the existing literature on individuals' methods of carrying out software maintenance, this paper fills the gap. In order to explore the role of political processes by individuals, this paper draws on qualitative studies of two small information systems organizations and specifically, we tried to answer our research question: How software maintainers in small organizations develop individual heuristics?

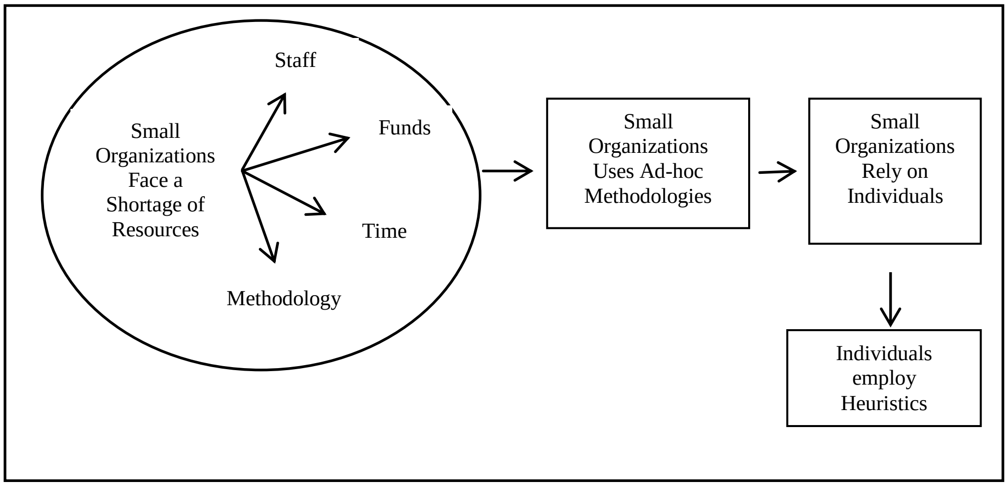
The rest of the paper is as follows. First, we discuss existing literature, then present relevant literature review; next we explain our method and results and finally present contribution and future direction.

LITERATURE REVIEW

A review of literature regarding software maintenance projects in small organizations show a paucity of attention to small organizations' software maintenance efforts (Brodman and Johnson, 1994). While the importance of small organizations is found to be gaining ground in the process improvement related publications (Hofer, 2002; Aysolmaz, and Demirörs, 2011), work on software maintenance in small organizations is not prevalent (Anonymous, 2011). An investigation into the salient factors of software maintenance showed that certain elements are regarded as highly critical for software maintenance. These elements are methodologies (April et al, 2005), identification and implementation of changes (Pigoski, 1997), and available resources at the disposal of small organizations such as individuals, funds and tools (Jorgensen and Sjoberg, 2002). Existing software maintenance methodologies, however, show that they fail to provide a comprehensive solution to the needs of small organizations (April et al., 2005). It is because they fall out of the reach of small organizations, due to their high costs, lack of ease in implementation or their inability to provide a holistic solution. Hence no specific methodology was found to be helpful for small organizations; rather ad-hoc methods were found to be the norm among small shops (Grubb and Takang, 2003). Furthermore, it was learned that identification of changes occupies a huge amount of software maintainers' time (Lientz and Swanson, 1980). Identification of changes is achieved through techniques such as program comprehension, reverse engineering, defect analysis and defect prevention. Implementation of changes requires much planning and care. Given the limitations small organizations face, resources especially individuals were found to play a vital role in the success of software (Pfeffer, 1992).

Hence, as depicted in Figure 1 below, this literature review enabled us to conclude that small organization face a shortage of resources that include staff, funds, time and even methodologies, which lead to reliance on ad-hoc methods undertaken by the individuals. Individuals come out as the main drivers of small organizations' software maintenance efforts. They are the ones that weave together various threads of processes, identification and implementation of changes and utilization of appropriate resources that lead a given software maintenance project's success. These individuals employ certain heuristics to carry out their jobs.

Figure 1: Summary of Literature Review



Simon (1957) defined heuristics as adaptive strategies undertaken by individuals when their capacities are limited. While the occurrence of heuristics utilization may be more significant in small organizations, they are used by us all the time in all fields. Bray and Sniderman (1985) noted utilization of heuristics by citizen while making sense of politics. When a politician is Republican, voters infer that she would be strong on defense, would support low taxes and low government intervention. An investigation of individual’s utilization of heuristics reveals that such individuals are often considered experts in their fields. While explaining experts, Sniderman, Brody and Tetlock (1991) mentioned that “the comparative advantage of experts is not that they have a stupendous amount of knowledge, but that they know how to get the most out of the knowledge they possess.” In other words, not only do experts or experienced resources employ certain cognitive heuristics, but they are very much likely to employ them appropriately (Lau and Redlawsk, 2001).

A review of literature on how experts develop their expertise led us to look into the fields of Psychology and Cognitive Psychology. From there, we learned that in developing expertise, individuals employ some sort of thinking, decision making, and reusing information.

Related to thinking, three views were found. First is the view by Associationists, which goes back to Aristotle. According to this group individuals think in associations. Thought process follows chains of events and objects that are similar, fall in the same time, or are found in the same space. Hobbes and Locke, who are considered Associationists, explained unit of thinking as an association between two ideas (Humphrey, 1963). They stated that all human knowledge consists of either ideas or associations between ideas. The second view of thinking belongs to a group of scientists who oppose the point-of-view of Associationists. Otto Selz (1964) is one of them who stated that thinking does not represent linkages; rather they exhibit a process of filling gaps in a structural complex (knowledge). Yet a third view has been presented by Gestalt psychologists who explained thinking as a search process that relates one aspect of a problem situation to another, which results in a structural understanding. Thus “structural understanding," "overall insight," or "high-level understanding" of looking at all parts together to satisfy a goal (solve a problem) is what thinking is all about, according to Gestalts (Mayer, 1983).

Related to decision making, experts are found to be heavily involved in decision making. Hastie and Dawes (2001) related decision making to choice selection. They stated that decisions are needed if there is more than one choice available; each choice gets projected to take to certain outcome; and each outcome has consequences that can be assessed based on personal values and goals. Also, people make decisions based on habits, conformity to what others are doing, or based on religion/cultural/authority mandate. The process of decision making can be either rational or due to unconscious dominance or bad behavior due to a childhood trauma as explained in the Psychoanalytic Theory by Freud(Hastie and Dawes, 2001).

Related to reusing refers to repeating certain choices that have worked in the past. These assist experts in developing certain intuition. However it was noted that expertise is more than intuition only; it reflects acquired knowledge, experience and skills (Ericsson and Smith, 1991; Hunt, 2006).

Furthermore, heuristics acquisition or development depends on the approximate number of years in deliberate practice (experience) to attain a high level of expertise (Chase & Simon, 1973). In their research on expert decision making by Chess masters, it was noted that chess masters were able to quickly id best moves, whereas mediocre players did not even consider best moves. Chess masters do this due to their repertoire of 50k to 100k patterns that they recognize immediately. Strong players need a decade to collect such repertoire (Chase & Simon, 1973)
Yet another concept, Information Cascade, in which information gets pass down from others, is another source of heuristics formation. When people observe the actions of others and then make the same choice, regardless of their own information, information cascade occurs (Bikhchandani, Hirshleifer & Welch, 1992)

RESEARCH METHODOLOGY

Qualitative research approach was used to conduct empirical studies in two small I.T. organizations and five software maintenance projects. Collected data was converted into labels and concepts to generate categories. Resultant categories confirmed findings from literature that for software maintenance projects, small organizations face shortage of resources and rely on individual heuristics.

Data Collection

Case Study Methodology was utilized to collect data from information systems that are considered small organizations and from projects that were undertaken by these organizations.

Case Study Method is a widely used technique for carrying out qualitative research in information systems (Orlikowski and Baroudi, 1991). It is deemed as an appropriate method for a situation where “a how or why questions are asked about a contemporary set of events over which an investigator has little or no control” (Yin, 1994). Support for using case study can be found in the work of Benbasat, Goldstein and Mead (1987) also. They encourage using interpretive case study when the research and theoretical development understanding of the particular phenomenon are at a formative stage (Benbasat et al., 1987).

Data was collected from Interviews and existing documents. Interviews constituted semi-structured open ended questions and ranged from 40 – 60 minute time periods. All interviews were tape recorded and were fully transcribed. Reviewed documents included project plans, customer service requests, software application user guides and software specification documents. This collection provided good data about the involved actors, processes taking place in small organizations and details related to the dimension of size and success/failure of projects, software maintenance resources and tools, and enablers and inhibitors of software maintenance.
For data collection purposes, two sites were selected. Both are Information Systems Departments (ISD) of two U.S. Mid-Atlantic public universities. The first university has a user base of 20,000 students and 3,000 employees. The Information Systems Department under study has a total of sixteen staff employees including seven developers, five analysts, three managers and one director. The second university has a user base of 3,800 students and 826 employees. The Information Systems Department under study has a total of eight staff employee including four developers, two analysts, one managers and one director. A vast majority of work carried out in both information system departments is in Software Maintenance. Site data and organizational information is provided in **Table 1** below.

Table 1: Data Collection Sites

	Organization One	Organization Two
Director	1	1
Managers	3 (1)*	1 (1)*
Analysts	5 (2)	2
Software Developers	7 (2)	4 (2)
Department Staff Size	16	8
Majority of work in Software Development (SD) or Software Maintenance (SM)?	SM	SM
Department Type	I.S. Department (Administrative)	I.S. Department (Administrative)
Type of Business	University	University
User Base – Students	20,000	3,800
User Base – Staff	3,000	826
*Actual Interviews		

Data was collected in general about software maintenance practices and in particular about a few software maintenance projects that were carried out in these organizations. There were two sets of projects that were looked at. The first set included three projects with different sizes (large, medium and small sizes). The second set included two projects with different outcomes (a successful and a failed project). Data for this set was collected in a second round to data collection. This data provided us with a good way of refining the initial findings. Specifically, this data assisted in the Selective Coding phase of Grounded Theory Methodology. By Selective

Coding phase, a tentative core category is found. For us it was: Individuals utilize heuristics to get goals achieved in software maintenance projects in small organizations. At such a point in Grounded Theory development, Glasser (1978) recommends "theoretical sampling." Theoretical sampling is the deductive part of Grounded Theory that involves selectively sampling new data with the core in mind.

Data Analysis

Grounded Theory Method (GTM) was utilized for data analysis (Glasser and Strauss, 1967). For data analysis, Grounded Theory Method (GTM) was chosen due to its strengths in analyzing and interpreting data that is originated from real life phenomenon on one hand and is directly related with current theories on the subject matter. Strauss and Corbin described GTM as "a qualitative research method that uses a systematic set of procedures to develop an inductively derived grounded theory about a phenomenon"(Strauss and Corbin 1998). Specifically, we used GTM to identify software maintenance processes and generated findings in the abovementioned sites. These findings were then sensitized with existing research on heuristics related topics to present key factors that assist in formation of individual heuristics (core category). GTM enabled data analysis was conducted over three phases: (1) Open Coding, (2) Axial Coding and (3) Selective Coding.

In Open Coding, data was initially fractured and then examined line-by-line. Each line of collected data (interview transcripts) was reflected upon by writing memos and identifying labels, concepts and categories (Strauss and Corbin, 1998). Categories provided rich descriptions through the definition of properties. This is an iterative process of identifying concepts, categories and properties identification that continues till one is satisfied with the results.

In Axial Coding, the fractured data was reassembled and categories were related to their subcategories to form more precise and complete explanations about phenomena (Strauss and Corbin, 1998).

In the Selective Coding phase, the final story line emerged. It is the process of selecting core categories and systematically relating them to other categories (Strauss and Corbin 1998). Resultant core and related categories from Selective coding are presented in **Table 2**.

T**able 2**: Selective Coding Categories

Type	**Title**
Core Category	Individual Heuristics Formation Process
Category	Actors
Category	Experts
Category	Individual Heuristics

These categories were further augmented and facilitated by “Theoretical sensitivity” which implies being sensitive to, and drawing inspiration from, existing research while developing the empirically grounded theoretical model, as recommended by GTM scholars (e.g., Urquhart, Lehman and Myers, 2010; Glaser 1978). In particular, we were interested in developing an insight into the nature of the individual heuristics utilized in small organizations' software maintenance projects.

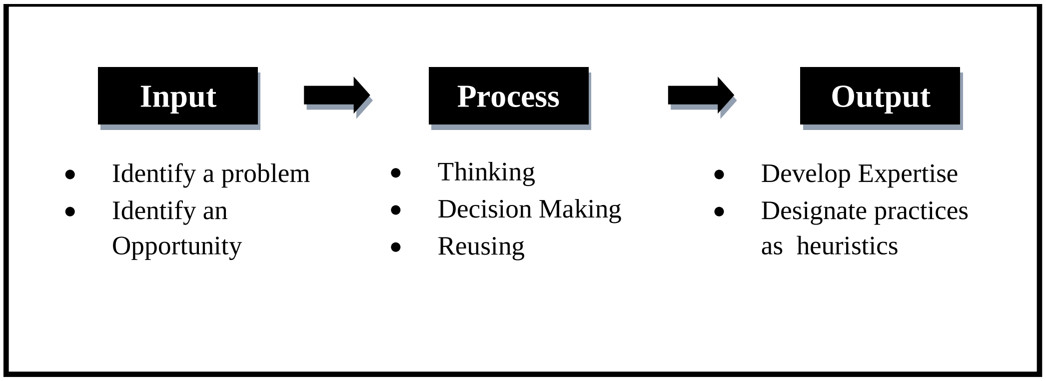
The core finding of this paper is in small organizations in order to perform various critical functions of software maintenance, individual actors, as the drivers of organizations, utilize certain heuristics which are formed over a formation process. These individuals are able to develop and form certain heuristics because of the experience and expertise that they have acquired over a long time period. While discussing experts, Sniderman, Brody and Tetolck, (1991) mentioned that the comparative advantage of experts is not that they have a stupendous amount of knowledge, but that they know how to get the most out of the knowledge they possess. In other words, not only do experts or experienced resources employ certain cognitive heuristics, but they are very much likely to employ them appropriately (Lau and Redlawsk, 2001).

RESULTS

As noted above in software maintenance projects, small organizations face challenges of lack of resources - personnel, time, funds and methodology (Pigoski, 1997). Such shortage of resources results in ad-hoc processes which cause extreme reliance on key individuals for the success of the project and the heuristics they have developed. In the absence of formal software maintenance guidelines, individuals in small organizations develop a pattern where they rely on their own rules gleaned from the experiences gained over the years (anonymous, 2011). These rules developed from ad-hoc processes are unwritten guidelines that have been used in the maintenance projects and represent heuristics that are followed in any new projects that are initiated. In other words, in small organization's software maintenance projects the vacuum created by formal guidelines is filled by individual heuristics.

From data review and analysis of empirical data, we have learned that individual heuristics are formed over a process of formation that begins with an individual looking at a problem or an opportunity, applying certain process, and generating output (results). If the results are good then the same rules/heuristics that were applied to the problem or opportunity are repeated. With repetition, these rules become individual heuristics. This process is presented as the Heuristic Formation Model in **Figure 2** below:

Figure 2: Individual Heuristics Formation Model



The three phases of heuristic formation that are found in existing small organizations' software maintenance projects are presented in detail below.

1. Input

This is the first phase of Individual Heuristic Formation Model. It begins when an individual working for a small organization identifies a problem or recognizes an opportunity for an enhancement. In one of the interviews, when asked how he begins solving a problem, an analyst responded:

> ***"the trigger is always either a problem that needs to be solved or an enhancement or an opportunity that others are availing that we are not. That gets us thinking about a solution"***

Thus a problem or an opportunity gets the ball rolling towards heuristics formation. It should be noted that input could be initiated by any of the stakeholders such as team members from the I.T. groups, user's community, executive management, or legislatures. How the input gets initiated is out of the scope for this paper. Once a problem or an opportunity reaches an individual, the heuristic formation model initiates. It was noted in data that while management assigns projects to individuals, details on how to carry them out are often not clearly communicated. This is especially true in small organizations where paucity of resources is the norm. Management and subordinates in small organizations in our study were found to wear multiple hats. Members of the management team were found to play the role of salesmen, economists, resource allocators, and strategists (thinking about the next project). One analyst noted

> ***"While it is true that we have little say in the kind of work we get, we often have a lot leeway in how we proceed with what is assigned to us. Often our managers are so short on resources that they cannot afford to work with us beyond getting the business. Often they get the business and drop it in our laps."***

While finding the first phase of heuristic formation model, we were trying to distinguish how such formation would be different in projects which are not software maintenance related. In other words how would projects be different if they were initiated in software development? Software Maintenance is defined in IEEE Standard 1219 as “the modification of a software product after delivery to correct faults, to improve performance or other attributes or to adapt the product to a modified environment” (IEEE, 1993). Given this definition, we found software maintenance projects would be no different in how heuristic formation. The only difference found were that related to the level of experience with particular situations or projects. This was mentioned by one of the developers.

> ***“With brand new projects, we have no clue how to start; with work that is built on existing software, we have some experience.”***

This showed that in with software maintenance work, individuals have some track record to work with.

2. Process

The second phase of the Individual Heuristic Formation Model is process. In our empirical investigation, this phase was seen to occur when a problem or an opportunity has already been triggered. This means an individual, i.e., an analyst, a developer or a manager in a small organization has received the message or a task to consider looking at a problem or an opportunity and he has started giving it attention. Reaching this stage requires utilization of resources, i.e. organizational time or funds. Management should have authorized adoption of the project and should have assignment of specific resources. One manager stated:

> ***“Before any work is assigned to a developer, we (management team) thoroughly review priorities and make assignments according to skills and availability of resources.”***

Once management assigns a project to an individual, the process stage commences. It should be noted that this commencement takes place in the mind of the software maintainer (an expert). As we were interested in the process of heuristic formation, we dug deep into how experts operated by analyzing the data and by getting clues from theories. As a result three components were identified as sub-phases of the process stage. They are thinking, decision making and reusing information. Each of these components is explained below. It should be noted that each of these sub-phases corresponded to the theoretical constructs related to experts and how they process information, as listed in the literature review above. More of this correspondence is explained below. Such connection between empirical evidence and theory is regarded as theoretical sensitivity and is highly recommended by grounded theorists such as Urquhart, Lehman and Myers (2010), Glaser (1978) and Suddaby (2006).

Thinking: The first sub-phase of the “Process” phase of the ‘Individual Heuristic Formation Model is ‘thinking.’ One of the managers explained how he thinks about a given software maintenance project.

> ***"I do my thinking by relating new problems to old problems, filling needs of customers and never forgetting the big picture"***

This quote epitomizes the three concepts that were found in the empirical data. These are relating ideas, filling needs and overall view. As presented in Table 3 below, these concepts are related to the three theoretical views mentioned in the literature review. They are the views of Associationists, opposing to Associationists and the Gestalt theorists (Mayer, 1983).

Table 3: Thinking Concepts

Empirical Data	**Literature Review**	**Main**
Relating ideas	Associationists	Aristotle, Humphrey (1963)
Filling needs	Opposing to Associationists	Otto Selz (Hastie and Dawes (2001)
Looking at the big picture	Gestalt theorists	(Mayer, 1983)

According to Associationists, which goes back to Aristotle, thinking relies on associations. Thought process follows chains of events and objects that are similar, fall in the same time, or are found in the same space. Hobbes and Locke, who are considered Associationists, explained unit of thinking as an association between two ideas (Humphrey, 1963). They stated that all human knowledge consists of either ideas or associations between ideas.
There are some scientists who oppose this point-of-view. Otto Selz, 1964, is one of them who stated that thinking does not represent linkages; rather they exhibit a process of filling gaps in a structural complex (knowledge).

Yet a third school thought represented by Gestalt psychologists explained thinking as search process that relates one aspect of a problem situation to another, which results in a structural understanding. Thus "structural understanding," "overall insight," or "high-level understanding" of looking at all parts together to satisfy a goal (solve a problem) is what thinking is all about, according to Gestalts (Mayer, 1983). In our empirical data, we found all of these three views of thinking in use by individuals while carrying out software maintenance efforts.

Decision making: The second sub-phase of the "Process" phase of the 'Individual Heuristic Formation Model' is 'Decision Making.' When we asked an analyst, how he goes about solving problems, here is what he said:

> ***"to make a decision, we rely on previous experience of something that has worked before, or we go to someone else who has worked on a similar problem to pick their brains and then we explore doing what they did and after weighing their consequences we make a decision."***

This implies that decision making is a key activity in heuristic formation. However, it closely relies on knowledge gained from similar experiences in the past with similar problems. The process of how decision making takes place has been explained by Hastie and Dawes (2001). They mentioned that a decision is needed if there is more than one choice available; in that situation a person reviews each choice, projects the outcome associated with it and assesses the outcome based on personal values or goals. This process was found in our empirical investigation as well. Mostly individuals began solving a problem or working on an opportunity and then proceeded to select among various choices. In most instances, those choices were selected that had worked for them before. Hence, experience of positive results played a big role in choice selection.

Reusing: The third sub-phase of the "Process" phase of the 'Individual Heuristic Formation Model' is 'Reuse of information.' While there is extensive literature available on reuse of information in software engineering, their focus has been on technical and organizational factors, largely ignoring cognitive characteristics of individuals. Despite anecdotal evidence that cognitive heuristics play a role in successful artifact reuse, few empirical studies have explored this relationship (Parsons & Saunders, 2004). People frequently reuse information. When people observe the actions of others and then make the same choice, regardless of their own information, information cascade occurs (Bikhchandani et al. 1992). Such reuse was found in full action in the small organizations, as found in this study. This was seen in form of reusing previous documented information, or picking the brains of existing senior staff. A manager mentioned:

> ***"Senior staff holds institutional memories that are often tapped for reuse. Without such knowledge, we would be reinventing the wheels all the time."***

While information reuse seems to be a frequent event in small organizations, certain biases should be watched for in decision making and information reuse. They are representativeness, availability, and anchoring-and-adjustment (Kahneman and Tversky, 1979; Webb and MacMillian, 1995). Representativeness refers to making an uncertainty judgment on the basis of the degree to which the new situation is similar in essential properties to its parent population. This should be watched as despite the similarities there might be stark differences between the two situations. Availability is another bias that can clog the information reuse. It refers to estimating frequency or probability by the ease with which instances or associations come to mind. Availability has been reported to be influenced by imaginability, familiarity, and vividness (Kahneman & Tversky, 1979). Anchoring-and-adjustment, the third bias, involves starting from an initial value that is adjusted to yield the final answer. The initial value, or starting point, may be suggested by the formulation of the problem, or it may be the result of a partial computation. In either case, adjustments are typically insufficient (Kahneman & Tversky, 1979). In other words, when people are given a problem for which there exists a correct solution and an initial estimate of its solution, they tend to provide final estimates close to the initial estimate when asked to solve the problem (Parsons and Saunders, 2004). This is termed anchoring. The anchoring helps humans simplify problem solving in a complex situation without conscious effort. However, since they are not aware they are using the heuristic, people often fail to make adequate modifications to an initial solution and produce estimates that are severely flawed with respect to the correct solution. This is termed an adjustment bias (Shanteau, 1989).
This discussion shows that reuse of information is a useful tool in heuristic formation; however, information re-use is susceptible to cognitive biases in their formulation and use. While the topic of biases came up in discussions with study participants, no meaningful assessments can be

made from collected data. Hence, biases are mentioned as precautionary note here; further empirical investigation is needed to fully understand its relationship to the heuristic formation model.

3. **Outpu*t***

This is the third major phase of Individual Heuristic Formation Model. This phase relates to how expertise gets developed and how repeated decisions eventually become individual heuristics and get utilized. As noted in the Individual Heuristics Formation Model (Figure 2), the output of heuristics gets developed by initially going through two phases of Input and Process. In other words the process begins by an individual looking at a problem or an opportunity, applying certain processing methods (thinking, decision making and reusing, and finally generating outputs (heuristics). If the results from applying heuristics are good then they are repeated.
A key finding from data analysis is heuristics are closely related to expertise. Thus, we found in this organization, the main sources of individual heuristics are individuals who have been working with an organization for a long time and have developed certain expertise that has worked for them repeatedly. On developer stated:

> ***"I know what to do in a given situation because I have certain levels of expertise that I have developed over years of experience, training and trial and errors."***

This expertise gives them a kind a of professional intuition. As noted before, however, expertise is more than intuitions though; it reflects acquired knowledge, experience and skills (Ericsson and Smith 1991; Hunt 2006; Yates and Tschirhart, 2006). Furthermore, heuristics acquisition or development depends on the approximate number of years in deliberate practice (experience) to attain a high level of expertise (Chase & Simon 1973). In their research on expert decision making by Chess masters, it was noted that chess masters were able to quickly id best moves, whereas mediocre players did not even consider best moves. Chess masters do this due to their repertoire of 50k to 100k patterns that they recognize immediately. Strong players need a decade to collect such repertoire (Chase & Simon 1973)

Yet, another element that distinguishes experts and lead to heuristics is the speed of decision making. Experts are able to rapidly make good decisions due to heuristics. Heuristics are principles or devices that contribute to reduction of search in problem-solving activity (Newell, Shaw and Simon 1958). Those practices and rules employed by individuals that succeed get reused by individuals and become individual heuristics.

DISCUSSION AND FUTURE RESEARCH PLANS

This paper presented a unique perspective on how individuals in small organizations carry out their roles in software maintenance projects. With results from empirical data in small I.T. organizations, it was found that individuals utilize heuristics that they have developed over years of experience after gaining certain expertise. These heuristics enable individuals to implement

solutions and to quickly make decisions conducive to the benefit of their organizations. How these heuristics are developed in existing organizations is explained in this paper with a model that depicts various stages of heuristic formation. We hope that such information will be useful to both Information Technology (IT) practitioners and researchers to better understand inner workings of software maintenance projects in small organizations. While our empirical study presented in this paper does not claim to have generalized findings, we do provide useful snapshots of how small organizations operate in real life context dealing with software maintenance issues and limited resources. Nonetheless, in the future, we plan to repeat a good number of such studies in multiple sites which can lead to emergence of general guidelines.
Moreover, in the future, we plan to conduct further research to uncover relationships of our heuristic formation model with other dimensions in small organizations and research findings by other scholars. For instance, the Individual's Heuristic Formation Model presented in this paper is strikingly similar to how computers operate. Both go through the phases of input, process and output as shown in **Table 4**.

Table 4: Humans and Computers Processing Information

Information Processes	**Humans**	**Computers**	**Heuristics Formation Model**
Input	Use external stimuli and five senses (Smell, Sound, See, Taste, Touch)	• Keyboard • Mouse	• Problem • Opportunity
Process	Use Temporary Memory	RAM	• Thinking • Decision Making • Reusing
	Use Permanent Memory	• Hard Drive • ROM	
	Brain makes decision	CPU	
Output	Communicate Decisions	• Printout • Monitor	• Develop Expertise • Designate practices as heuristics

Evidence of similarities of computers to humans in processing information was initially underscored in the cognitive revolution of the 1960s when computer programs were created to mimic human actions such as arithmetic and chess playing. Newell and Simon categorized humans and computers as both "information processing systems" (Newell and Simon, 1972). As noted in **Table 4** above, Hastie and Davies (2001) stated that many aspects of human thinking including judgment and decision making can be produced in computational models. Both humans and computers deal with the three phases (input, process and output) in the same way. Inputs are communicated through external stimuli and are sent in for processing via sensory media such as sight, sound, smell, taste and touch. Received data is then processed in temporary memory (similar to computer's Random Access Memory - RAM) and then are sent to the brain of a human or a Central Processing Unit of a computer. Brain / CPU is relied upon to glean established rules and syntax from information stored in long term memory (hard drive and ROM). Ultimately both entities make a decision and communicate it through an output medium. Such flow is similar to what was presented in this paper's individual heuristics model. We hope to conduct additional empirical investigations in finding similar relations between this model and other existing research frameworks.

REFERENCES

1. Anquetil, N., De Oliveira, K. M., De Sousa, K. D. and Batista Dias, M. G. (2007). Software maintenance seen as a knowledge management issue. *Information & Software Technology, 49*(5), 515-529.

2. April, A., Hayes, J. H., Abran, A. and Dumke, R. (2005). Software maintenance maturity model (SM^{mm}): The software maintenance process model. *Journal of Software Maintenance & Evolution: Research & Practice, 17*(3), 197-223.

3. Aysolmaz, B and Demirörs, O. (2011). A detailed software process improvement methodology: BG-SPI. Vol. 172, Part 3, 97-108, DOI: 10.1007/978-3-642-22206-1_9

4. Benbasat, I., Goldstein, D.K., Mead, M. (1987). The case research strategy in studies of information systems, *MIS Quarterly*, Vol. 11, No. 3 (Sept), pp. 369-386

5. Benestad, H. C., Anda, B., and Arisholm, E. (2009). Understanding software maintenance and evolution by analyzing individual changes: A literature review. *Journal of Software Maintenance & Evolution: Research & Practice, 21*(6), 349-378.

6. Bikhchandani, S., Hirshleifer and Welch, I (1992). A theory of fads, fashion, custom, and cultural change in informational cascades. *Journal of Political Economy,* 100(5), 992-1026

7. Bray, H.E., and Sniderman, P.M.(1985). Attitude attribution: A group basis for political reasoning, *American Political Science Review* 79:1061-1078

8. Brodman, J.G. & Johnson, D.L. (1994). Businesses and Small Organizations say about CMM? *Procedings of the 16th International Conference on Software Engineering,* Sorrento, Italy

9. Chase, W.G. and Simon, H.A.(1973). The mind's eye in chess. In W.G. Chase (Ed.) Visual information Processing , New York: Academic Press

10. Ericsson, K. A. and J. E. Smith (1991). Toward a general theory of expertise: Prospects and limits. Cambridge, England, Cambridge University Press.

11. Glaser B.G. (1978). Theoretical Sensitivity, Sociology Press, Mills Valley, CA.

12. Glaser, B.G., and Strauss, A.L. (1967). The discovery of grounded theory. Chicago, IL: Aldine.

13. Grubb P., and Takang, A.A. (2003). Software Maintenance Concepts and Practice, 2nd Ed. World Scientific Publishing, Singapore.

14. Hastie, R. and Dawes, R. (2001). Rational choice in an uncertain world: the psychology of judgment and decision making, Sage Publications, Thousand Oaks, CA

15. Hofer, C. (2002). Software development in Austria: Results of an empirical study among small and very small enterprises, *Proceedings Euromicro Conference, 2002*

16. Humphrey, G (1963). Thinking: An introduction to its experimental psychology. New York: Wiley

17. Hunt, E. (2006). Expertise, Talent and Social Encouragement. The Cambridge Handbook of Expertise and Expert Performance. K. A. Ericsson, N. Charness, P. J. Feltovich and R. R. Hoffman. Cambridge, Cambridge University Press.

18. IEEE, Std. 1044, 1993. IEEE standard classification for software anomalies, IEEE.

19. Jørgensen, M., and Sjøberg, D. I. K. (2002). Impact of experience on maintenance skills. *Journal of Software Maintenance & Evolution: Research & Practice, 14*(2), 123-146.

20. Kahneman, D., and Tversky, A. (1973). On the psychology of prediction, *Psychological Review*, pp.237-251.

21. Kahneman, D., and Tversky, A. (1979). Intuitive prediction: Biases and corrective procedures, TIMS Studies in Management Sciences pp. 313-327.

22. Ko, A. J., Myers, B. A., Coblenz, M. J., and Htet, H. A. (2006). An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks. *IEEE Transactions on Software Engineering, 32*(12), 971-987.

23. Lau, R. R., and Redlawsk, D. P. (2001). Advantages and disadvantages of cognitive heuristics in political decision making. American Journal of Political (add detail) Vol. 45, No. 4, October

24. Lehman, M.M. (1966). The Programming Process. IBM Res. Rep. RC 2722, IBM Research Center, Yorktown Heights, NY 10594

25. Lientz B. P., Swanson E. B. (1980). Software Maintenance Management. Addison Wesley, Reading, MA

26. Mayer, R (1983). Thinking, Problem Solving, Cognition, W.H. Freeman and Company, New York

27. Newell, A., and Simon, H.A. (1972). Human problem solving. Englewood Cliffs, NJ: Prentice Hall

28. Newell, A., Shaw, J.C. and Simon H.A.(1958). The processes of creative thinking, The RAND Corporation Paper, P-1320

29. Orlikowski, W. and Baroudi, J.(1991). Studying information technology in organizations: Research approaches and assumptions, *Information Systems Research* 2.1, 1-28.

30. Parsons, J., and Saunders, C. (2004). Cognitive heuristics in software engineering: Applying and extending anchoring and adjustment to artifact reuse. *IEEE Transactions on Software Engineering*, 30(12), 873-888.

31. Pfeffer, J. (1992). Understanding Power in Organizations, *California Management Review* Winter; 34, 2; pg. 29

32. Pigoski TM. (1997). Practical software maintenance: Best practice for managing your software investment. Wiley, New York NY, 29-36, 117-138.

33. Poon, S. (2000). Business environment and internet commerce benefit—a small business perspective. European Journal of Information Systems, Volume 9, Number 2, 1 June 2000 , pp. 72-81(10)

34. Shanteau, J. (1989). Cognitive heuristics and biases in behavioral auditing: Review, comments and observations. *Accounting, Organizations & Society*, 14(1/2), 165-177.

35. Simon, H. A. (1957). Models of man, social and rational: Mathematical essays on rational human behavior. New York: Wiley.

36. Sniderman, P.M., Brody, R.A, and Tetlock, P.E. (1991). Reasoning and Choice: Explorations in Political Psychology. New York: Cambridge University Press

37. Software Industry Statistics for 1991-2005, Enterprise Ireland, 2006; www.nsd.ie/htm/ssii/stat.htm

38. Strauss, A and Corbin, J.(1998). Basics of qualitative research: Techniques and procedures for developing grounded theory, 2nd Ed., Sage Publications, Thousand Oaks, CA

39. Suddaby, R. (2006). "From the Editors: What Grounded Theory is Not," Academy of Management Journal (49) 4, pp. 633-642.

40. Swanson, E. B., and Dans, E. (2000). System life expectancy and the maintenance effort: Exploring their equilibration. *MIS Quarterly, 24*(2), 277-297.

41. Takang, A.A., and Grubb, P.A. (1996). Software Maintenance Concepts and Practice. Thompson Computer Press London, UK.

42. Urquhart, C. (2001) An Encounter with Grounded Theory in E. M. Trauth (Ed.) Qualitative Research in IS: Issues and Trends, Hershey, PA: Idea Group Publishing, pp. 104 -140

43. Webb, S and MacMillian, J. (1995). Cognitive Bias in Software Engineering, *Comm. ACM*, vol. 38, no. 6, pp. 57-69

44. Yates, F.J. and Tschirhart (2006). Decision-Making Expertise. The Cambridge Handbook of Expertise and Expert Performance. K. A. Ericsson, N. Charness, P. J. Feltovich and R. Hoffman. Cambridge, Cambridge University Press.

45. Yin, R.(1994). Case study research design and methods, 2nd Ed., Sage Publications, 9.