AC 2012-4544: INOCULATING NOVICE SOFTWARE DESIGNERS WITH EXPERT DESIGN STRATEGIES

David R. Wright, North Carolina State University

David Wright earned his Ph.D. in computer science from North Carolina State University. He is currently a Research Associate in the Computer Science Department, overseeing the day-to-day operations of four different research projects. Wright has taught a variety of undergraduate courses at NCSU and other local institutions. His research interests include software design and engineering education, focusing on ways to help students think more like engineering professionals than students, as well as developing teaching and learning tools and strategies that help keep students interested in computer science.

©American Society for Engineering Education, 2012
Inoculating Novice Software Designers with Expert Design Strategies

Introduction

Well-developed design decision-making skills are important for competent software developers, yet it has been noted that formal education is generally inadequate for developing these abilities in novices. Boulangera and Smith note that industry needs engineers and designers that can be immediately and productively effective. Studies of industry expectations of new computer science and software engineering graduate skills highlight a number of deficiencies in the preparation of students for professional careers. Among the most commonly noted gaps between expectations and actual skills are

- the ability to understand software systems as different than single-user programs;\textsuperscript{6,51}
- the ability to visualize different perspectives or views on a software system;\textsuperscript{10,11}
- the ability to think critically and reflectively;\textsuperscript{31,38}
- systems analysis and design skills;\textsuperscript{6,31,51} and
- problem-solving and investigative skills.\textsuperscript{6,10,11,31}

As more and more of our world becomes dependent upon computer-based systems, future software developers and designers must develop effective decision-making skills and strategies in addition to the technical knowledge they acquire through their education.

The decision-making strategies commonly taught in Computer Science/Software Engineering curricula and employed by novice software developers/designers are not effective when applied to large, complex real-world problems. Student (novice) designers tend to think linearly and concentrate only on the problem at hand. Novices also tend to use trial and error strategies and lack confidence in their design decisions. The production of artifacts (e.g., programs, UML diagrams, etc.) often takes precedence over the design decision-making processes used to create these artifacts. The problems students are asked to solve are usually compact, well-formed, and have a small set of “correct” solutions. Performance is measured by how closely the student’s work match the desired outcomes and specifications. How the student developed the artifacts is generally not a concern: if the “correct” solution was delivered, the student must have used the “correct” design approach.
While these skills may be sufficient for translating thoroughly documented designs into working code, they are inadequate for reasoning about the large and complex software systems that the modern world depends upon.\textsuperscript{24} When faced with large and ill-defined problems, the linear and deductive software design approaches commonly presented in software design and engineering textbooks break down.\textsuperscript{18,39} Students of software design and engineering need tools to help them learn, develop, and apply expert design strategies to help them mature into productive software design professionals.

The Principles, Patterns, and Process Framework ($P^3F$) has been developed in response to this need. The $P^3F$ is a unique synthesis of results and ideas from a wide range of disciplines that provides novice software designers with a simple yet rich and flexible guide to help them quickly cultivate the key processes and behaviors of expert designers. This paper presents a summary of a research study undertaken to validate the usefulness and effectiveness of the $P^3F$, answering the overarching research question: \textit{How can software design students develop effective design decision-making strategies like those used by expert designers?} In order to answer this question, additional research questions also must be answered, including:

- Are software design students able to demonstrate the expert strategies incorporated into the framework when presented with a complex, ill-structured design problem?
- Does the framework improve the quality of design artifacts produced by the student?
- Does the framework improve the student’s perception of his or her design abilities and enable them to be more confident in their design decision-making?

This paper presents the results of an interrupted time-series experiment conducted to answer this question. Data collected included subjects’ self-assessments of their design behaviors, verbal protocols of the subjects working on complex software system design problems, and the artifacts representing their solutions to these problems. The self-assessments were compared against the behaviors observed in the recorded protocols to characterize the subjects’ design behavior before and after the introduction of the $P^3F$ and identify changes attributable to their use of the Framework. The solution artifacts were evaluated against a design quality rubric and these results were analyzed to identify improvements in the quality of the solutions that correlated with the introduction of the $P^3F$. The primary data for this study was the observed behaviors of the subjects, with the self-assessments and artifact quality evaluations serving to corroborate or question the observational data.

The results of this study suggest that the Principles, Patterns, and Process Framework can help novice software designers adopt expert design strategies and improve their personal design skills. One of the five subjects completing the study demonstrated significant transformations in their construction and navigation of their view of the design problem and evolving solution as well as an improvement in the quality of the solutions they produced. Another two subjects adopted, to a lesser extent, aspects of the expert strategies incorporated in the $P^3F$ while retaining much of their novice behaviors. The other two subjects showed little or no behavioral change or difference in solution artifact quality.

The next section reviews the background and prior work that provide the foundation for this study. An overview of the Principles, Patterns, and Process Framework is then presented, followed
by a discussion of the experimental design and implementation. The analysis techniques and experimental results are then introduced and the paper finishes with the conclusions drawn from this study and ideas for future research building upon these results.

**Background and Prior Work**

The body of research about learning to design software systems is scarce, with most research efforts studying expert designers and/or comparing novices and experts. Intending to develop cognitive models of design behaviors, these studies consider novices and experts to be oppositional entities and do not attempt to explain how a novice develops the skills and knowledge to become an expert. This situation is not unique to software design research: studies of experts and novices in engineering design and architecture also fail to address the learning processes associated with developing design expertise. Other than identifying and describing behaviors of expert and/or novice designers, such studies offer little advice to aid in the transition from novice to expert. However, the prior research into expert and novice design behaviors provides a rich and varied set of data that characterizes the common design behaviors and thinking strategies of both novices and experts.

A review of the literature identified three broad strategic areas that have been common subjects of study or highlighted in the results of studies of both expert and novice designers. How a designer views the design problem at hand is intimately connected to how the designer attempts to find a solution to the problem. Equally important is how designers manage their own knowledge of the design problem and use it to manipulate and navigate through the solution they are developing. The third strategic area identified deals with the way the designer makes design decisions, including what information is used and how the decisions are grounded or justified.

**Expert Design Strategies**

The strategies, behaviors, and processes used by experienced designers have been frequent subjects of study. The principal goal of these studies has been to understand expert strategies in order to build models that can be used to implement “expert systems” to support the work of designers in specific fields. From these studies, three common strategies emerge. First, expert designers are able to maintain a systemic or “big picture” view and understanding of the design problem within the application space where the problem occurs is a critical skill exhibited by expert designers.\(^{20,25}\) In order to maintain the big picture view of the system, expert designers make extensive use of abstractions, mental models, and simulations.\(^{2,12,40}\)

The second common expert design strategy is the ability to navigate effectively through the application problem space and the developing solution. Several key skills expert designers use to navigate large and complex system designs include a high tolerance for ambiguity and uncertainty,\(^{25,52}\) the ability to “chunk” particular elements or aspects of a system for closer attention,\(^{55}\) and problem setting. Problem setting is a dynamic cognitive strategy that allows the designer to expand, contract, and reorient his or her focus within the design space.\(^{58}\)

The third common strategy employed by expert designers is the ability to confidently make design decisions, even when faced with incomplete information. Expert designers are characterized by open-mindedness, a high tolerance for ambiguity, orientation to purpose, and a preference for ‘soft’ or subjective information.\(^{52}\) Expert designers rely on a variety of knowledge sources when making design decisions, one of the most important being “first principles.”\(^{20}\) While background and domain knowledge are important aspects of design decision-making, their use tends to be
imprecise and directed towards framing and structuring the problem rather than directly making decisions about a particular solution.  

Rather than rigidly applying a single methodology or approach, the expert designer remains flexible and adaptive, opportunistically using the information at hand to refine the solution and development path. Intermediate forms and hypotheses represent the designer’s knowledge at a particular point in time and also expose new aspects of the problem and solution. Inflexible design methodologies can severely constrain the designer’s ability to adapt or change their approach when faced with new information and understanding.

Novice Design Strategies

Students are generally good problem-solvers but poor decision-makers. Exposure to expert strategies and decision-making processes has the potential to enhance and complement other types of design education. However, the emphasis of engineering design education is on algorithmic problem-solving skills while practitioners deal with complex, ill-defined, and open-ended problems that require decision-making skills and strategies that can contend with ambiguous, uncertain, and poorly-defined information. The problems students are asked to solve are usually compact, well-formed, and have a small set of “correct” solutions. Such problems lend themselves to logical, analytical solution development, and are much more easily graded for correctness and performance than open-ended problems with a large set of satisfactory solutions. As a result, students learn what they need to to achieve their desired grades. While developing these analytical problem-solving skills may be sufficient for completing typical academic work, they are inadequate for preparing the future software developer for professional practice. When faced with large and ill-defined problems, the linear and deductive software design approaches commonly presented in software design and engineering textbooks break down.

Novice designers view design problems much differently than their expert counterparts. In particular, novice designers have a strong tendency to think linearly and concentrate only on the problem at hand. This problem-centric view coupled with a deductive problem-solving approach effectively places blinders on the novice designer. Eckerdal and her colleagues suggest that this kind of behavior should be taken in context, since students are generally “trained” to solve problems rather than to design solutions. Novices also tend to focus on the literal or surface features of a problem in order to fit what is known and what is unknown about the problem into their fact-based knowledge. The novice’s literal focus also exhibits itself as a rush to an embodiment or implementation because the novice cannot separate concepts and ideas from the implementation of those ideas. This emphasis on the obvious and literal aspects of a design problem appear to be a symptom of a deeper deficiency in the skill set of novice designers: they are unaware of the information they need most of the time.

A superficial view of the design problem coupled with a lack of attention to the environment in which the solution will function is a frequent cause of errors and failures. This superficial view of a design problem contributes to the novice attitude that understanding and formulating the problem is of little importance and that the problem statement contains all of the information necessary to construct a complete solution. The shallow view of design problems typical of novices affects how they manage and navigate the accumulating knowledge about the problem and solution. One of the consequences of their superficial perspective is evidenced by their difficulty with or failure to scope the design problem. Comparing novices and experts in architectural
design, Casakin reports that novices’ inability to identify and apply analogies in their design processes hindered their ability to identify constraints not explicitly stated in the design problem.\textsuperscript{17} Likewise, novices often fail to identify important goals and objectives that are foundational to resolving design problems.\textsuperscript{49}

Novice designers’ primary search and navigation method is top-down, depth-first decomposition of the problem.\textsuperscript{39} This constricted strategy is further hampered by difficulties in integrating the decomposed pieces back into a complete and working solution.\textsuperscript{47} When decomposition fails the novice designer, the common response is to apply a trial-and-error strategy, exhaustively searching through possible solutions until a working candidate is found.\textsuperscript{4,5,49} Trial-and-error is not an efficient design strategy, particularly when large numbers of possible solutions are generated and implemented without an overarching plan or goal.\textsuperscript{17,27} Further aggravating this kind of weak and ineffectual design strategy is the failure to critically evaluate proposed solutions and hypotheses about the design problem and generally inconsistent design problem-solving approaches.\textsuperscript{2,42}

Very little research could be found that specifically examines how novices make design decisions. Novices tend to follow linear, deductive problem-solving approaches that may be effective for well-formed problems, but that are inadequate for ill-structured design problems.\textsuperscript{18,39} Unlike experts, novices attempt to avoid uncertainty and ambiguity, and seek a solid, unchanging foundation for their decisions whenever possible.\textsuperscript{8,59} Their literal view of the problem and attempt to fit it into their fact-based experience is a means of reducing this uncertainty and ambiguity.\textsuperscript{13,44} When they cannot eliminate uncertainty, novice designers lack confidence in their design decisions.\textsuperscript{4,5} Once a decision is made, novice designers find it difficult to revisit or reconsider the decision, particularly when such reconsideration would entail reversing it.\textsuperscript{14} Novices’ “rush to implementation” is further exemplified by their expressed need to document their decisions in code rather than through other means of communication.\textsuperscript{33,43} Because they strive to base their decisions on what they believe to be literal or unchangeable facts and these decisions are often made concrete through implementation, novices are generally unable or unwilling to change decisions that have been made.\textsuperscript{59}

**Overview of the $P^3F$**

The Principles, Patterns, and Process Framework is made up of four related elements: a set of fundamental design principles, pattern structures that help “chunk” information into manageable and relatable pieces, a process for applying the design principles and navigating through the developing design, and a template for making informed and confident design decisions. Together these elements provide a simple and extensible set of tools that incorporate the expert design strategies noted above in a structure and format that can be used by novice software designers.

Pattern structures are foundational to the $P^3F$ because they express both rules for creating structures and the relationships between structures in space and time.\textsuperscript{19} They also provide a means for descriptively chunking information into manageable units referenced by a metaphoric name.\textsuperscript{50} Software design students should be familiar with design patterns as they are a recommended part of the undergraduate curriculum as well as commonly used in professional practice.\textsuperscript{1}

The Design Principles Pattern Language uses a common design pattern format to express and document the fifteen fundamental design principles in the $P^3F$.\textsuperscript{67} Use of the pattern structure provides supplemental contextual and relational information that helps explain how, when, where, and why each principle is applicable. In addition to providing a basis for design decision-making,
these principles also provide an abstract structure for the developing system to help facilitate navigation through the design space through the use of the pattern structure.

The process element of the $P^3F$ is based upon Alexander’s *Fundamental Differentiating Process.* This process provides a template for identifying and iterating over generative design decision sequences that allow a system’s structure, behavior, and interaction with its environment to unfold as a coherent whole that smoothly integrates with its environment. With each new iteration, the student is directed to refresh the view of the system as a whole, incorporating the changes made in the previous iteration. The student is also directed to evaluate the current system using the design principles as guide to identify the strengths and weaknesses of the system, identifying the part of the system that need the most immediate attention.

The final element of the $P^3F$ is a template for making and recording design decision that we call the *Decision Pattern.* This template, also expressed in a pattern format, abstracts and structures the key elements necessary to make a critically-informed decision. Requiring students to use a strict format for documenting their decisions can provide insights into how the student arrived at a particular solution. This is similar to showing each step in a mathematical proof or derivation: a step represents the application of some operation or transformation intended to move the problem closer to a successful resolution. The template also provides the student with a tool for reflecting on a sequence of decisions leading up to a particular point in the design of a software system.

The results of this pilot study suggest that after being introduced to the $P^3F$, students’ tendencies to engage in behaviors characteristic of novices declined, and their tendency to exhibit expert behaviors increased. This suggests that this sample of students were able to understand and use the learning tools in the $P^3F$, and that when they did, their work took on aspects of expert designers’ strategies.

**Study Design and Implementation**

This research study was implemented as a interrupted time-series quasi-experiment with four observations conducted at two to three week intervals. Each observation session included the completion of a design skills self-assessment questionnaire and solving a complex, open-ended software system design problem. The $P^3F$ was introduced during the second session after the subject completed the design problem and was reviewed at the start of the third session. During the problem-solving part of the sessions, the researcher acted as both the observer and as a representative of the client described in the design problem. The subjects were audio-video recorded while working on the problems and encouraged to verbalize their thoughts in the process. The artifacts produced by the subjects to represent and describe their proposed solutions were also collected for analysis. This section briefly summarizes the experimental design that was implemented for this study.

At the start of each observation session, subjects were asked to complete a questionnaire to assess their own design abilities and skills. The purpose of the self-assessment questionnaire is to identify changes in the subjects’ perceptions of their own design skills and strategies over the course of the experiment. To quantify these perceptions, the questionnaire asked subjects to indicate their agreement or disagreement with statements describing their strategies and behaviors while working on complex software design problems. The statements were constructed based on the design behaviors and strategies of experts and novices identified in the literature review and pilot study. The statements were grouped based on the three general behavior types described in the
literature: viewing the design problem, navigating the problem/solution space, and making design decisions and were developed into phrases completing a common sentence fragment.

Several key requirements for the design problems used in this study were determined before engaging in a search for the problems. The problems should be realistic and non-trivial, drawn from variety of application domains and business settings, and should be described in business terms rather than a list of explicit requirements and specifications. The problems should also have a degree of ambiguity created by missing information, conflicting statements, and/or extraneous comments. Finally, all of the problems should have approximately the same level of difficulty and complexity. Eight problems were chosen from a variety of systems analysis and design and software engineering textbooks for use in the study through an iterative refinement and pruning process. An example of the problems chosen is shown in Figure 1.

---

**ABC, Inc. Payroll System**

You have been tasked with designing a batch payroll system for ABC, Inc. The system will consist of a database of the company’s employees, and their associated data (such as time cards). The system must pay all employees the correct amount, on time, by the method specified. Various deductions must be taken from employees’ gross pay, and records of deductions and other payroll-related information must be maintained to support periodic reporting of payroll information to local, state, and federal tax authorities and for the employee’s tax records. Following are notes taken during an interview with the customer.

- Some employees work by the hour. They are paid an hourly rate that is one of the fields in the employee record. They submit daily time cards that record the date and number of hours worked. Hourly workers are paid weekly, with the payroll period starting on Friday morning and ending the following Thursday night. They are paid every Friday afternoon for the week ending the night before. If an hourly worker works more than 40 hours in a weekly payroll period, they are paid 1.5 times their normal rate for those extra hours.

- Some employees are paid a flat salary. They are paid on the last working day of the month. Their monthly salary is one of the fields in their employee record.

- Some salaried employees are also paid a commission based on their sales. They submit sales receipts that record the date and the amount of the sale. Their commission rate is a field in their employee record. Commissions are paid every other Friday afternoon for the 2-week sales period ending the night before (Thursday).

- Employees can select their method of payment. They may have their paychecks mailed to the postal address of their choice, have the checks held by the paymaster for pickup, or request that paychecks be directly deposited into the bank account of their choice.

- Some employees belong to a union. Their employee record has a field for the weekly dues rate. Their dues must be deducted from their pay. Also, the union may assess service charges against individual union members from time to time. These service charges are submitted by the union on a weekly basis and must be deducted from the appropriate employee’s next pay amount.

- The payroll application will run once each working day and pay the appropriate employees on that day. The system will be told what date the employees are to be paid to, so it will generate payment records from the last time the employee was paid up to the specified date.

---

Figure 1: Study problem example

The target population for research subjects was advanced undergraduate computer science students, specifically students who had completed the junior-level software engineering course. This requirement was made because these students would have had some exposure to large software systems as well as to software development methodologies used in professional practice, and would
be less likely to be overwhelmed by the complexity of the design problems. Due to a lack of responses from this target group, the participation requirements were relaxed to include graduate students in the computer science program and undergraduates who had successfully completed all of the prerequisites for the software engineering course. Seven subjects began the study and five completed all of the study activities.

**Data Analysis**

The three different types of data collected in this study, questionnaire responses, verbal protocols (as audio/video recordings), and artifacts produced as solution representations, required three distinct analysis methods. This data and methodological triangulation is an important aspect of this research, as it enhances the rigor of the work, reduces internal and external threats to validity, and helps reduce the risk of inappropriate certainty. Rather than attempting to perform detailed statistical analysis, the data was examined subject-by-subject to explore the details of the data in a qualitative manner.

The self-assessment questionnaire was designed to measure how the research subjects perceive their own design skills and strategies with the goal of identifying changes in those perceptions that might be associated with their exposure to the $P^3 F$. The questionnaire contained statements describing strategies and behaviors exhibited by novices and experts in previous studies of designers. The statements were grouped into three categories: viewing the design problem, navigating the design space, and making design decisions. Subjects indicated their level of agreement with each statement using the scale 1) Strongly Disagree, 2) Disagree, 3) Not Sure, 4) Agree, 5) Strongly Agree. The statements are not oppositional, so that strong disagreement with a statement describing novice behavior does not imply an expert-like tendency or vice-versa.

In verbal protocol analysis, the coding scheme is a mapping between the recording of what a subject says and does while performing some task and the cognitive processes of interest to the researcher. In this study, the cognitive processes of interest are the design decision-making processes of the research subjects, specifically how subjects make their decisions and why they make the choices they do. Based on this, the coding scheme was designed to identify and describe design decisions in the recorded protocols. For this study, a coding scheme using four variables was designed to capture information about design decisions. The variables used are:

**Level** - The level of abstraction or detail within the design of the system where the decision is made.

**Type** - The type of design decision made.

**Rationale** - The influences and/or forces that were the basis for the decision and the choice made.

**Subject** - The subject or object of the decision and any alternative solutions identified.

Two individuals with extensive experience coding verbal and textual protocols were hired to perform the coding for this study. To help ensure objective and reproducible coding, neither coder had any connection or interest in the outcome of this research. To assess the consistency and reliability of the coding process, Fleiss’ kappa was calculated to measure the inter-rater reliability between the two coders. The overall κ statistic for all codes and protocols was computed to be...
0.52, with values for individual codes ranging from 0.43 to 0.89. The coded protocols were then analyzed using an iterative approach to identify and classify the subjects’ observed behaviors and the order in which they occurred. The analysis process consisted of five phases as shown in Figure 2. The resulting three-level codings were then compared across the four observations for each subject to identify any behavioral changes that occurred in the course of the study. Subjects’ behaviors were also compared with their self-assessment responses and with the evaluations of their solution artifacts to produce an overall view of their performance.

The third data set collected was the artifacts produced by the subjects as representations of their solutions. An exhaustive search of the literature did not identify a compact, general-purpose set of criteria for measuring the quality of a software system design. Software quality rubrics do exist, but they are specific to a particular perspective or aspect of a system, such as its performance, security, reliability, or usability. Expanding the search to other engineering and design disciplines yielded the Design Quality Rubric used to measure design quality in a study of students in a mechanical engineering capstone design course\textsuperscript{60}. Two additional design quality categories were added to tailor the criteria more specifically to software systems: soundness and scalability. The resulting rubric was reviewed by the external evaluators, both of whom suggested the addition of a “security” category. The final software design quality rubric used to evaluate the design artifacts contained eight metrics is shown in Table 1. The subjects’ artifacts were anonymized and presented to independent evaluators without any indication of their chronological order. The evaluators were given the associated design problem statements to use as a reference in their evaluation of the solutions. Cohen’s kappa statistic was calculated to measure the inter-rater reliability of the artifact evaluations. The overall $\kappa$ statistic for all metrics and artifacts was computed to be 0.89, with values for individual metrics ranging from 0.81 to 0.94. Overall, these $\kappa$ values suggest close agreement between the two evaluators\textsuperscript{45} which was reinforced by their similar comments associated with their artifact evaluations.

Subject A Results
<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>The design meets the technical criteria and customer requirements</td>
</tr>
<tr>
<td>Feasibility</td>
<td>The design is feasible in its application and implementation</td>
</tr>
<tr>
<td>Soundness</td>
<td>The design is sound and reasonable in its use of programming languages, algorithms, data structures, and existing technology</td>
</tr>
<tr>
<td>Creativity</td>
<td>The design incorporates original and novel ideas, non-intuitive approaches, or innovative solutions</td>
</tr>
<tr>
<td>Simplicity</td>
<td>The design is simple, practical, usable, reliable, maintainable, and safe, and avoiding any approaches that would introduce unnecessary complexity into the system implementation</td>
</tr>
<tr>
<td>Scalability</td>
<td>The design is sufficient to efficiently meet reasonably expectable increases in demand and/or data volume without modification</td>
</tr>
<tr>
<td>Security</td>
<td>The design incorporates elements sufficient to protect against security threats that would commonly be associates with the type of application and business described in the problem statement</td>
</tr>
<tr>
<td>Overall</td>
<td>Overall impression of the design solution</td>
</tr>
</tbody>
</table>

Subject A was female, a junior Computer Science major who had not yet taken the undergraduate Software Engineering course, and was not taking the class during her participation in this study. She had no professional experience in software development and had not taken any advanced Computer Science elective courses.

Subject A’s responses to the self assessment questionnaires were very consistent across all four observation sessions, exhibiting a strong association with novice behaviors. In particular, the subject saw herself as implementation-oriented, preferring to express a design in executable code rather than more abstract design symbolism. Subject A also expressed an inclination to fit problems to known solutions, a strong desire for detailed, explicit information, and very little tolerance for uncertainty or ambiguity. Subject A also disagreed or strongly disagreed with nine of fifteen expert-oriented statements on the questionnaire, alternating between agreement and disagreement on the others across the four sessions.

In the first two problem-solving sessions, Subject A exhibited many behaviors associated with novice designers. In particular, the subject relied heavily on explicit information given in the problem statement, making little or no effort to gather additional information, commenting that “the customer would have included more information if it was needed to design the system” on several occasions. Subject A also maintained a very narrow focus on the particular part of the problem under consideration, decomposing that part to try to identify concrete details of the solution. The subject’s overall approach to solving both problems was disorganized and sporadic as she jumped from one part of the problem to another.

During the third and fourth sessions, after the introduction of the $P^3F$, Subject A demonstrated several behaviors that are more commonly associated with expert designers, and very different than her behavior during the first two sessions. The subject exhibited a more organized approach to solving both problems, verbally describing her understanding of the problem and developing solution...
at regular intervals during the sessions. In the third session, Subject A also began intentionally revisiting prior design decisions when new information became available. Although she still relied on the explicit details given in the problem statement, the third session also saw her asking more questions and attempting to gather more information outside that statement. These behaviors continued throughout the fourth session, with the subject expressing and revising a plan for solving the problem, and actually following that plan as she worked on her solution. Another significant change was the use of a breadth-first approach to information gathering and design decision-making as opposed to the depth-first approach that dominated the first two sessions.

To document her solutions, Subject A chose to create diagrams and notes by hand rather than using any of the electronic tools provided. She explained that she had little or no experience using those tools, and found it easier just to draw and sketch by hand. The subject’s sketches were a combination of simple UML class diagrams and finite state machine bubble diagrams, and showed little change in quality over the four observations. The evaluators remarked that the artifacts produced by Subject A were superficial and confusing, and did not communicate an understanding of the problem sufficient to construct a solution.

Subject A’s self-assessments and design quality evaluations did not indicate any effects that might be attributable to the introduction of the $P^3F$, although the analysis of the verbal protocols suggests there might be some small effect reflected in the subject’s observed behaviors. Although she continued to rely on the explicit information given in the problem statement, she also seemed much more willing to ask questions to gather additional information after the introduction of the framework. In the third and fourth observation sessions, Subject A was also more organized in her approach to the design problems and tried to construct and maintain a big-picture view of the problems and solutions. The $P^3F$ process is stated in a step-by-step fashion and incorporates steps that ask the designer to consider the context and environment of design decisions, and this construction was intended to make it easily accessible and applicable for novice users. While Subject A did not follow the process in an explicitly observable manner during the last two sessions, the iterative revisiting of her view of the system as a whole and a more organized strategy for approaching the problem suggest that she was trying to use the process in her work on the problems.

Comparing the self-assessment results with the observed behaviors also revealed similarities between Subject A’s perceptions of her design strategies and the strategies she used while working on a design problem. In her self assessments, the subject consistently agreed with several statements indicating novice-like design thinking. The responses to these statements suggest that the subject considered herself to be implementation oriented, tries to fit problems to known solutions, desires explicit and concrete information, and relies on personal experience. These behaviors were very evident during the first two observation sessions, corroborating the self-assessment results. During the last two sessions, these behaviors were still present but to a lesser degree than in the first two sessions, although this was not reflected in the self-assessment responses. This suggests that she was consciously trying to use aspects of the $P^3F$, but that she did not have enough experience with the framework to internalize it to the point of altering her self perceptions.

The artifacts Subject A produced for evaluation were not consistent with the verbal descriptions made while she was constructing the artifacts. In particular, the subject verbally described her solution in clearer detail and with better organization than what she put on paper. This inconsistency remained constant over all four sessions and the resulting artifacts, suggesting that this subject
might be better at describing what she is thinking than she is in documenting it in some explicit form, particularly when time is limited.

Overall, the results for this subject suggest a modest move towards more expert-like behavior when designing solutions to complex software system problems. The observed change may have been due only to a conscious effort to apply the most accessible part of the $P^3F$ for the benefit of the study, but because the subject did not make specific references to the framework it is difficult to determine if that was indeed the case. The observed behaviors also confirmed several aspects of Subject A's novice-oriented view of herself.

**Subject B Results**

Subject B was male, a junior Computer Science major who had not yet taken CSC326 (Software Engineering), and was not taking the class during his participation in this study. He had no professional experience in software development and had not taken any advanced Computer Science elective courses.

Subject B agreed or strongly agreed with eleven of sixteen novice-oriented self-assessment statements across all four observation sessions. The subject characterized himself as needing to focus on one issue at a time with a strong desire to restructure problems to fit solutions within the realm of his personal experience. His responses to novice-oriented statements also reflected a strong desire for explicit and specific detailed information about the problem and a need to avoid ambiguity and uncertainty. The subject’s responses to expert-oriented statements seems to contradict this self-assessment, as the subject also characterized himself as open to uncertainty with a preference to use abstractions and metaphors as placeholders for details that may be filled in later. There was no evidence of consistent change in the subject’s self-assessment of his design skills occurring as a result of the introduction of the $P^3F$.

During all four of the observation sessions, Subject B relied heavily on his own experiences and on concrete details explicitly stated in the problem description. He also worked hard to restructure the problem to fit a solution he was familiar with, regardless of whether or not this solution was the best one for the situation. The subject’s strong focus on a single issue at a time precluded the development of an integrated view of the system.

Subject B was confident in his decisions, but this confidence was often manifested in an arrogant or domineering manner. The subject’s confidence rested on his apparent presumption that because he was the “expert” in the conversation, he knew what was best, and his proposals should not be questioned. He was also willing to change directions at will while designing, but his behavior during the observations seemed haphazard and seemed to employ a trial-and-error approach. The lack of serious and sustained attention to any part of the problem resulted in shallow and superficial solutions that demonstrated a lack of attention to detail.

None of the results accumulated for Subject B suggest that the introduction of the $P^3F$ had any significant effect on the subject’s design strategies. His self-assessment responses indicated a consistent novice-like view of himself, including characteristics such as a narrow focus, the desire to restructure problem to fit with known solutions, a strong reliance on personal experience and explicit information. The subject also responded that he perceives himself to be a confident designer, willing to revisit and change decisions made earlier in a design process. The results of the verbal protocol analyses and the design quality evaluations suggest that his strategies are more novice-oriented than expert-like.
The subject’s behaviors, solution quality, and self-assessment analyses do not support a conclusion that the introduction of the $P^3F$ triggered any changes in Subject B. The subject was not observed to exhibit an increase in expert-like behaviors or a decrease in novice characteristics, and his perceptions of his own design skills and strategies did not change in a significant way. Furthermore, while the quality of the solution produced for the third problem was a significant improvement over the evaluation of the second-session artifact, the scores for the second solution were much lower than the first. Only two of the third artifact metrics scored higher than the first session, so the increase in quality from session two to session three was more like a recovery to the initial quality level for most of the metrics.

**Subject C Results**

Subject C was a male second-year graduate student in the Master of Computer Science program. He had one year of work experience in software development, and had taken many advanced computer science courses at the undergraduate and graduate levels. The subject’s responses to the novice-oriented statements on the self-assessment questionnaire were consistent agree or strongly agree for thirteen of the fifteen statements. Overall, these responses indicated an orientation towards concrete, functional details in problem statements, restructuring problems to fit familiar solutions, and a high level of uncertainty about the quality of the proposed solution until realized in executable code. Subject C’s responses to the expert-oriented statements in the questionnaire contradicted many of his responses to the novice-oriented statements. In particular, the subject indicated strong agreement with statements concerning the use of metaphors and abstractions and a high tolerance for ambiguity and uncertainty.

Throughout all four observations, Subject C demonstrated an orientation towards concrete, implementation-level aspects of the problems he worked on. He not only assumed that the problem statement contained all of the information he would need to solve the problem, he also asserted that any other source of information was not important. This included information directly provided by the “customer.” Taking this view, the subject avoided any uncertainty, as his solutions were directly defined by the problem statement, and if there were any flaws or limitations in his solution they were a product of an incomplete statement, which he could not do anything about.

Subject C’s solution evaluations were, at best, fair or average, and predominantly poor. The improvements shown between sessions two and three were not significantly sustained through the fourth session artifacts. The evaluators commented on the shallow and superficial nature of his solutions to all four of the problems, also noting that the subject did not appear to make any attempt to identify critical issues described in the problem statements. Overall, these results do not provide evidence for any changes in performance, behavior, or self-perception that could be attributed to the introduction of the $P^3F$.

Subject C maintained a strong novice orientation across all four observations sessions, as demonstrated by his self-assessments, observed behaviors, and quality evaluations. In the self-assessments, the subject consistently agreed with statements indicating a desire for fixed, concrete information, uncertainty about his decision-making unless he could see code execute properly, and a general orientation towards an implementation-based design.

These perceptions were supported by the behaviors he exhibited during all four observation sessions. Subject C relied heavily on the explicit information provided in the problem statements as well as his own personal experience and knowledge. His problem-solving approach was linear and
driven by the order of topics in the problem statements. The subject avoided uncertainty at almost any cost, often by simply ignoring ambiguous or poorly structured parts of the problem.

Subject D Results

Subject D was male, a junior Computer Science major who was taking the undergraduate Software Engineering course during his participation in this study. He had one year of professional experience in software development and had not taken any advanced Computer Science elective courses prior to participating in this study.

Subject D’s self assessments suggested changes in his perceptions of his design strategies and behaviors. At the first two sessions, the subject indicated strong agreement with using a narrowly-focused approach that relied heavily on explicit, concrete information given in the problem statement as a supplement to his personal experience. At the third and fourth sessions, the subject’s responses indicated less agreement and disagreement, respectively, with these novice-oriented statements. Subject D’s first two self assessments also indicated disagreement with expert-oriented statements involving maintaining a system-level view of the problem and solution, using a flexible and opportunistic approach to problem solving, and using first principles as a basis for design decision-making. His responses to these statements also changed to agreement and strong agreement at the third and fourth sessions.

During the first two problem-solving sessions, Subject D used a trial and error approach to generating alternative solutions. He also spent much of his time gathering information without trying to assemble parts of it together into potential solution elements, a common characteristic novices employ when faced with complex and ill-formed problems. During the third session, after the introduction of the $P^3F$, Subject D took an iterative approach to gathering information and generating solution elements, explicitly using several of the principles given in the $P^3F$. He also regularly returned to reflect on his progress with respect to his understanding of the overall problem and customer goals. In the fourth session, the subject continued to build on the new behaviors and strategies that emerged during the previous observation and adding new strategies such as defining boundaries and relationships between elements of the problem and his proposed solution.

The evaluations of the solution artifacts produced by Subject D during the first two observations were in the “poor” to “average” range, and both evaluators commented on an overemphasis on specific hardware details in the proposed solutions. The evaluators also commented on the lack of attention to critical details of the problem in these artifacts. The solutions produced during the third and fourth sessions were rated average to very good. The evaluators noted evidence of the subject’s holistic and system-wide perspectives, as well as the identification of critical aspects of the problem that were not explicitly stated. They also noted that the subject documented different alternatives that were viable based on the information available, and the mature manner in which the various tradeoffs involved were considered and evaluated. Both evaluators also commented on the higher-level thinking that was evidenced in the artifacts produced during the third and fourth sessions.

The overall view of the results for Subject D suggest that the introduction of the $P^3F$ did have a positive effect on the subject, increasing his expert-like behaviors and decreasing his novice attributes. Compared to the other four subjects in this study, Subject D was working at a more expert-like level prior to the start of the study. This was reflected in his self-assessment responses
where he consistently asserted trying to maintain a system-level view of the problem and solution in
context and the ability to apply a flexible and opportunistic approach to solving complex problems.

These perceptions were substantiated by his observed behaviors. Subject D did demonstrate
several expert-like behaviors during the first two observation sessions, in particular the ability to
accept and manage uncertain and ambiguous information through the use of abstractions and limited
mental models. During the third and fourth sessions, the subject made several references to \( P^3 F \)
principles and demonstrated his ability to use them effectively as design tools. This analysis strongly
suggests that the introduction of the \( P^3 F \) to this subject was the principal cause of the changes in
his behavior, self-perception, and design performance.

**Subject E Results**

Subject E was male, a junior Computer Science major who was taking the undergraduate
Software Engineering course during his participation in this study. He had no professional experience
in software development and had not taken any advanced Computer Science elective courses.

Subject E’s self assessment responses indicated little or no change in his attitudes towards his
software design skills and behaviors over the course of the study. His responses were also somewhat
contradictory. For example, he agreed or strongly agreed with statements indicating he prefers
concrete information with specific details leading directly to executable code, while also indicating
that he easily uses abstractions and metaphors to create mental models of potential solutions. He
also indicated that his problem-solving method of choice was to use a trial and error approach with a
predefined plan for addressing parts of a problem, but agreed with statements describing his design
method as opportunistic with only a high-level plan. Subject E characterized himself as strongly
logical and deductive as well as creative and often creating solutions without a logical basis in the
information at hand.

During all four of the observations, Subject E demonstrated a very shallow and superficial
approach to generating solutions. He did not attempt to establish relationships between different
parts of the problems or between different elements of his solutions. Generally, he tried to fit the
problems into solutions he was very familiar with and dealt with uncertainty and ambiguity by
simply ignoring what he did not know or understand. During the fourth session, however, the subject
began using the contrast between what he did and did not know to help drive his problem-solving
approach. He also tried to build and maintain a system-level view of the problem and how his
solution could resolve various aspects of it at a high level. As part of this system-level view, Subject
E also used abstractions and metaphors as placeholders for elements that he did not yet understand,
revisiting these elements as he gathered new information to refine and flesh out details.

This change was also evident to the evaluators through the solution artifacts Subject E produced.
The artifacts from the first three sessions received poor to very poor evaluations, accompanied
by comments that the evaluators felt the subject was confused by the problems and produced
incomplete and very superficial solutions. The solution produced for the fourth problem received
average to very good evaluations. The evaluators commented that this solution indicated a good
understanding of the problem, including critical aspects that were not clearly stated. They also
remarked that the subject seemed to have developed an accurate mental picture of the problem, and
his proposed solution addressed nearly all of the key goals and objectives.

Viewed as a whole, Subject E’s self-assessments, observed behaviors, and artifact quality
evaluations present a mixed and often paradoxical picture of this subject. Several of his self-assessment responses were at odds with each other. For example, he agreed with statements expressing a need for concrete and detailed information, while also agreeing with statements indicating that he can easily handle uncertainty and ambiguity. He also responded affirmatively to preferring to use a deductive approach to problem solving, but asserted that he is most productive when he is creative and his decisions do not follow logically from the information at hand.

During all four of the observation sessions, his behavior demonstrated a shallow, narrow focus and a strong desire to avoid uncertainty and ambiguity, often by ignoring parts of a problem that he felt were not sufficiently well-defined for him to work on. The evaluations of the solution artifacts from this subject’s first three sessions confirmed these behaviors, with evaluator comments suggesting that the designer seemed confused and did not understand the problem he was trying to solve. Subject E’s solution for the fourth session, however, was considered by the evaluators to be well thought-out, demonstrating a good understanding of the problem with a holistic perspective.

In the fourth session, the subject was observed to create and maintain a big-picture view of the problem and solution, but the solution he described verbally still appeared to be very shallow. Comparing the actual solution artifacts to the transcript of the session, it was clear that the subject was devoting much more communication effort to his handwritten artifacts than he was to the verbalization of his thoughts. Overall, this suggests that the introduction of the $P^3F$ may have helped Subject E behave more like an expert in the third and fourth sessions. However, it is difficult to state this with authority because the changes noted by the evaluators were not clearly evident to the same extent in the subject’s self-assessments or in his observed behaviors.

**Discussion of overall results**

Triangulating research methodologies and their associated data collection and analysis techniques is especially important for early-stage research such as this study, since it helps to reduce risks and threats to the validity of the research as well as enhancing the rigor of the work. Of particular interest to the current work, these three distinct sets of data reduce the risk of misunderstood certainty by providing different views that together can provide stronger corroboration than any one of them alone. This corroboration between data sets was a factor in the analysis of two of the subjects in this study, and the lack of clear confirmation between different data leaves those results in a less than certain state.

The results for Subject D provide the most convincing evidence that the $P^3F$ can help novice software designers adopt and apply expert-like design strategies. After he was introduced to the framework, this subject’s perceptions of his own design strategies reflected an increased awareness and ability related to several important expert behaviors. Among these were the ability to consider multiple issues at one time and to use that information to construct multi-level and multi-perspective models of design alternatives. He also asserted an increased reliance on fundamental design principles and domain-centric best practices in his design thinking processes. These expert-like strategies were likely contributors to the increased confidence he reported having in his design decisions.

Subject D’s behavior, as observed through the four study sessions, reinforced these self-perceptions. After the introduction of the $P^3F$, the subject demonstrated an increased awareness of the relationships between different parts of the design problems and his developing solutions, and the ability to define and reinforce these relationships. He also acted more iteratively and reflectively
than he did in the first two sessions, frequently revisiting and evaluating previous decisions in light of new information. As part of this iterative and reflective process, he also maintained a clear vision of the context in which his proposed solution would eventually operate.

Finally, the evaluators commented on what they perceived as higher-level thinking processes in his solutions to the problems worked after the $P^3F$ was introduced. As evidence for these perceptions, they commented on his holistic and system-level perspectives on the problems, his consideration of novel solutions (including off-the-shelf software), and his awareness and evaluation of tradeoffs present in the alternatives under consideration. Although the quality scores for the fourth problem were not as high as those for the third one, the evaluators’ comments depicted more expert-like behavior in the fourth artifact compared to the third.

Two subjects showed some changes that might be attributable to their exposure to the $P^3F$, although in both cases the changes were only evident in one of the three data sources. Subject A’s responses to the self-assessment questionnaire remained strongly novice-oriented at all four observation sessions, and the quality of her solutions was consistently low. However, during the third and fourth sessions, after she was introduced to the $P^3F$, she was significantly more organized in her approach to solving the problems. Subject A also tried to establish and refine relationships between different parts of the problems and solutions, something that she did not do at all during the first two sessions. She also made reference to several of the principles presented in the $P^3F$, indicating that she was at least trying to use the framework.

Similarly, Subject E’s self-assessments remained consistently novice-oriented across all four sessions. His observed behavior also appeared to be superficial, highly dependent upon his own experience and the explicit details provided in the problem statements, and averse to uncertainty and ambiguity through the study. The solution artifact evaluators painted a different picture with their scores and comments. They suggested that the subject seemed to be confused during the first two sessions, producing solutions that were very shallow and incomplete. The artifacts from the last two sessions, however, demonstrated a good understanding of the problem, a holistic view of the big picture, and a strong grasp of the relationships between the different parts of his proposed solution.

Subjects B and C showed little to no evidence that they were affected by the introduction of the $P^3F$ in any way. Both of these subjects continued to demonstrate a strong reliance on personal experience and knowledge, a need for specific, detailed, and unambiguous information, and a desire to deliver implementation-level details that they could execute to evaluate. The evaluators also noted that the artifacts produced by both of these subjects were superficial, lacking attention to details stated in the problem description, and making no attempt to identify and respond to critical issues also mentioned in the problem statement.

**Conclusions and Future Work**

The research questions identified in the Introduction of this paper are answered by the results presented in the previous section. One of the subjects (D) was clearly able to demonstrate expert strategies and behaviors beyond those exhibited in the first two sessions after he was introduced to the $P^3F$, based on the analysis of the verbal protocols. The verbal protocol analysis also showed that Subject A displayed expert-like behavior to a limited extent in the third and fourth sessions, and made an clear effort to apply some of the $P^3F$ concepts as she worked on the last two problems. Subject E demonstrated expert-like strategies only during the fourth session. The analysis of the
data from Subject B and C did not provide clear evidence of expert-like behavior or strategies in any of their work on the design problems.

The answer to the question, “Are software design students able to demonstrate the expert strategies incorporated into the framework when presented with a complex, ill-structured design problem?” is a conditional yes. After being introduced to the $P^3F$, one subject was clearly able to demonstrate some of the expert strategies incorporated into the framework. Two other subjects also demonstrated expert strategies after their exposure, but to a more limited and less conclusive extent. The results of the verbal protocol analysis of the other two subjects design sessions showed no significant change in their behavior after the introduction of the $P^3F$ compared to their behavior in the first two observations.

The second question asks “Does the framework improve the quality of design artifacts produced by the student?” The answer to this question is positive but less conclusive than the previous question. Based solely on the evaluation scores, several of the subjects did show improvement in the quality metrics of the artifacts produced during the third observation session over those produced for the second session, prior to the introduction of the $P^3F$. However, the scores for these subjects’ second session artifacts were generally much lower than for the first session, and the third session scores were close to the first sessions scores. Also, the scores for the session 4 artifacts tended to show little change or a drop from the session 3 artifact evaluations.

If the evaluators’ comments are taken into consideration, the answer to this question is a bit more compelling. Comments regarding the artifacts produced by subjects D and E for sessions 3 and 4 indicate that the evaluators saw evidence of better fundamental designs, even though their evaluation scores did not reflect a higher quality rating. These comments also suggest that the lower scores reflect less complete designs, but that they had higher expectations that, given more time, these subjects would have produced even better proposals. In this light, the overall results of the design quality evaluations suggest that the introduction of the $P^3F$ can help to improve the quality of the designs produced by novice designers. However, the results also suggest that improvements in quality may lag changes in behavior. I would not consider this to be unusual, since there is almost always a “learning curve” associated with cultivating new skills, and it takes time and practice to become adept and effective.

“Does the framework improve the student’s perception of his or her design abilities and enable them to be more confident in their design decision-making?” is the final research question to be answered by this study. As with question 6, the answer to this question is yes for one of the subjects, and doubtful for the other four. Subject D’s responses to the self-assessment questionnaire indicate a decline in agreement with several common novice strategies and behaviors, and an increase in agreement with several expert characteristics. Most notable among these was that the subject’s responses indicate an increased reliance on fundamental design principles and best practices, and a decreased dependence on personal experience. Subject D’s responses also indicated an increased level of confidence in his design decision-making abilities.

One of the questions that must be asked of any research project is to what degree can the results and conclusions be generalized to the larger population represented by the study sample. In the case of this study, the sample population is too small to be a reliable indicator of general trends in the overall population of novice software designers. Furthermore, only one of the subjects (the graduate student) had the educational background originally defined as a prerequisite for participation in
the study, in particular an exposure to developing large, complex software systems as is generally covered in an undergraduate software engineering course.

As a result, no claims of generalizability are made for this research. That does not, however, eliminate the value that this study provides. The results suggest that the framework conclusively helped at least one of the subjects act and perform more like an expert than he did before he was introduced to the $P^3F$, and that this introduction also altered his perceptions of his own design skills and abilities.

The most significant limitation of this work was the size of the subject population. With only five participants, the ability to validly generalize the results of the study was effectively nullified. The small number of subjects was due to several factors, but the most critical one was the timing of the study and subject recruiting. The initial plan for the study intended it to be conducted during a fall semester, with subject recruitment starting the week before the start of classes for the semester. Complications arising from obtaining funds to compensate participants and the necessary Institutional Review Board (IRB) approval for human studies research delayed the ability to begin recruiting subjects until the end of the fall semester. These efforts did not garner any commitments for participation, and additional recruiting was delayed again until the start of the spring semester. Pushing the study into the spring added additional recruiting complications, as most of the target subjects were taking the senior design course as well as preparing for graduation and entering the professional workforce, and could not commit to the time required for participating in the study. As a result, none of the actual participants in the study fit the “ideal” subject for which the study was designed.

As an exploratory study, this research should serve as a foundation for additional studies, both to replicate and corroborate this work and to extend it by focusing on interesting and/or contradictory results. Other areas of inquiry that are derivative of this research include investigating any long-term effects of the $P^3F$ on this study’s subjects, revising the framework in response to feedback from the study participants, developing different versions of the framework to suit different levels of design experience and different design-related disciplines, and investigating potential non-design applications of the framework.

A replication study is a vital follow-up to the current work, particularly one that would involve a larger subject population over a longer duration. This would provide the opportunity to corroborate this exploratory work and strengthen the case for bringing the framework into the classroom, either as a dedicated course or as an integral part of one or more design-related courses. A follow-up study is also planned to investigate what, if any, long term effects the $P^3F$ may have had on the participants in this study.

Several colleagues have also expressed interest in other applications of the $P^3F$, and discussions are ongoing to explore these possibilities. The most immediately promising of these are the use of simplified elements of the framework in introductory-level programming classes and the development of a first-year “design studio” type of course for computer science and software engineering students. One of my dissertation committee members remarked that introducing new computer science students to the holistic and reflective thinking processes incorporated into the $P^3F$ may be more important to their ultimate success than waiting until they have been exposed to large, complex systems.

Other design-related disciplines that I believe could benefit from a “customized” version of the
include fields with a strong visual and/or tactile emphasis such as graphic design and visual arts, architecture, and creative writing. Non-design fields such as economics, project management, organizational behavior, and complex systems also intrigue me, and I occasionally have glimpses of how the framework could be used to gain a greater understanding of phenomena in these areas. One other application of the $P^3F$ that I would like to explore is the development of metrics for software systems based on the framework principles. I expect this to be a challenging endeavor because it represents an attempt to quantify the abstract and somewhat subjective values embodied in the principles.

While there is a small but significant body of research studying expert and novice design behaviors, there have been few attempts to bridge the gap to cultivate expert-like behaviors in novice software designers. This work contributes such an attempt combined with an effort to measure the change induced by the introduction of the $P^3F$. Furthermore, because the design of the Framework drew from a broad range of design-intensive disciplines, the $P^3F$ can be adapted to suit a variety of design environments. The positive results obtained in this study invite further investigation and refinement of the $P^3F$ in the realm of engineering design.

References


51. OUTLAY, C. N., AND KRISHNAN, P. Skill gaps and careers in is compliance: implications for is degree programs in the u.s. In *Proceedings of the 2010 Special Interest Group on Management Information System’s 48th annual conference on Computer personnel research on Computer personnel research* (New York, NY, USA, 2010), SIGMIS-CPR ’10, ACM, pp. 130–135.


55. ROBILLARD, P. N. The role of knowledge in software development. *Communications of the ACM* 42, 1 (Jan 1999), 87–92.


59. SIMMONDS, R. Limitations in the decision strategies of design students. Design Studies 1, 6 (1980), 358–364.


