Putting the Context in Context-Driven Testing (an Application of Cultural Historical Activity Theory)

Rebecca L. Fiedler
Indiana State University
Curriculum, Instruction, & Media Technology Department
200 N. 7th Street, Terre Haute, IN USA
1-812-237-2957
fiedler@indstate.edu

Cem Kaner
Florida Institute of Technology
Department of Computer Sciences
150 W. University, Melbourne, FL USA
1-321-674-7137
kaner@cs.fit.edu

ABSTRACT
Cultural Historical Activity Theory (CHAT) (Engeström, 1987) is a model that can organize systems-level thinking and analysis. In particular, CHAT focuses on the context in which an activity takes place. Therefore, it is particularly useful for context-driven software analysis. This paper introduces and explains CHAT, illustrates how it can be used in the field, and suggests ways to apply it to a variety of testing problems.

Categories and Subject Descriptors
D.2.5 [**Testing and Debugging**]:

General Terms
Design, Verification, Human Factors, Theory

Keywords
Context-driven Testing, Cultural Historical Activity Theory, Qualitative Methods, Requirements Analysis, Scenario Testing

1. INTRODUCTION
Software operates in a human system. It does tasks that humans want done. The program itself is a communication among several humans and a computer. The people will use the program, hold each other accountable for actions they take or information they obtain from the program, fix or enhance it, teach other people about it, and revise other systems to work with it. The computer will execute the program, carrying out a set of instructions in a language simplified so much that a mindless machine can follow them.

The people who are affected by the success of the software are its stakeholders. The pool of stakeholders is typically very diverse, with conflicting interests, even if we restrict attention to those stakeholders who work in the same company and profit from successful development of the software.

The discipline by which we empirically evaluate software in order to provide stakeholders with information about the quality of the software is called software testing.

The scope of testing is as broad as our pool of stakeholders is diverse. The scope of testing includes any of the ways in which the software might cause damage or loss, irritate or constrain stakeholders, or cause conflicts among stakeholders.

To understand the breadth of scope of software-related risks (the scope of testing and failure reporting), testers sometimes study systems theory (Laszlo, 1996; Von Bertalanffy, 1976; Weinberg, 1991, 2001).

Cultural Historical Activity Theory (Engeström, 1987), which we will often abbreviate as *CHAT* or *Activity Theory*, provides a structure for systems thinking that brings out the context in which a human activity is done such as the context(s) in which software is developed and used. CHAT has been repeatedly applied to software development in human-computer interaction and computer-supported cooperative work (Halverson, 2002; Kuutti, 1995, 1996; Kuutti & Arvonen, 1992; Nardi & Engestrom, 1998; Nardi & Redmiles, 2002), especially as these fields have begun to shift from user-centered design to context-centered design (Gay & Hembrooke, 2004; Kaptelinin & Nardi, 2006). When we talk about context-driven software testing (Kaner, 2009; Kaner, Bach, & Pettichord, 2001), the types of information that we consider "context" are the types of information that CHAT helps us identify and evaluate.

In the first part of the paper, we introduce CHAT using a simple example. The second part of the paper applies CHAT techniques and analysis to an extended analysis of software use drawn from Fiedler's dissertation. In the final section, we discuss ways to apply CHAT to testing.

2. INTRODUCING CHAT PRINCIPLES

Among activity theorists, the notion of "doing something" involves context and purpose. In other words, the actors in an activity are "doing in order to transform something" (Engeström, 1987). In developing this model of activity, Engeström (1987) sought to develop a model that was "the smallest and most simple unit that still preserves the essential unity and integral quality behind any human activity" (p.67). This model includes the collection of actors, actions, artifacts, and related activities that surround human endeavors. It also incorporates the sociocultural influences and, applied longitudinally, captures changes over time.

The CHAT model is depicted in Figure 1. According to Engeström's CHAT (1987), the *subject* acts on an *object* in order to achieve some kind of *outcome*. In other words, activity is purposeful.

- The *subject* may be an individual or a group. Who you choose to focus on, as the subject, determines the perspective or point of view for the analysis. As an Activity-Theory analysis evolves, the researcher or analyst might consider the same activity system through the eyes of several different subjects, such as a programmer, a tester, a project manager, an end user, and so on.
- An *object* can be raw materials being transformed into something else; concepts one must learn; or a 'problem space' at which an activity is directed (Center for Activity Theory and Developmental Work Research, n.d.). An object can be a thing (such as a computer) or a concept (if you studied recursion in a programming class, recursion was the object) or something the subject is doing in order to achieve the outcome (such as exercising to lose weight or going to a bar to make new friends). The object can be, and often is, shared with others who may also be working on the same object. That object carries the purpose or motive for an activity. There can be multiple purposes for an object and those purposes can evolve over time (Foot, 2002). For example, Sally is a software tester working on Company X's Hot New Product. Hot New Product is the object of Sally's attention. She shares this object with the programmers assigned to the project; the product manager trying to guide the development team to complete Hot New Product on time and under budget; and the marketing people working to promote Hot New Product to potential customers. The purpose or motive that we consider for the work on Hot New Product will change depending on the perspective we adopt. Some of the people might be working on Hot New Product because it's part of a medical device that has the chance to improve the quality of life for a loved one while others are working to collect a paycheck they will use to pay their bills and fund their hobbies. Adopting a corporate perspective, the motive might be to make a profit or to establish reputation in a new market.
- One or more *tools* mediate the subject's actions. In this model, *tools* mediate the interaction between subject and object. For example, the subject might use a tool to create, manipulate, or understand the object. These tools can be tangible (as in a hammer or a computer), symbolic (language or icon), or psychological (mental models or heuristics). The subject likely uses a tool to interact with the object.
- Human activity generally takes place in a c*ommunity* or group of individuals who share the same object such as colleagues in the work place, one's family, or even a group of hobbyists.
- The interaction of the subject with the community is typically governed by *rules, norms, and conventions*. These may be formal (part of the system), informal (quirky adaptations by a group), or technical (policies, laws, or mandates).
- Finally, the intersection of community and object is labeled *division of labor* as the community engages in the activity. The labor can be divided both horizontally and vertically. Horizontal division of labor is the division across members of a community with approximately the same status while vertical division of refers to tasks divided across different divisions of power.

One of the first steps in using CHAT is to consider the "unit of analysis." Decisions about the unit of analysis are driven by what one is interested in examining as well as the size and scope of the project in which one is involved. Engeström (2000) has used the CHAT framework for activities as large as the health care system in Finland and as small as individual doctor-patient interactions.

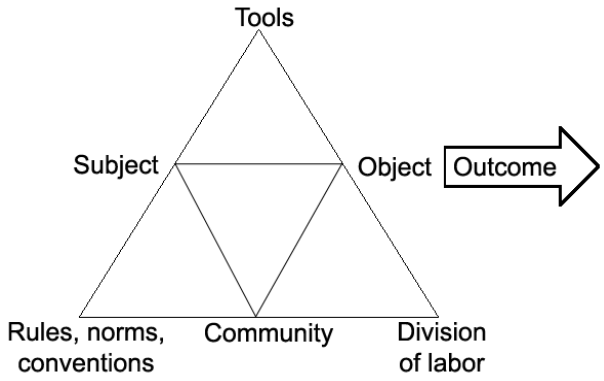

Figure 1. Engeström's CHAT Model

2.1 Russell's Grocery List Example

To introduce the application of CHAT, we will use Russell's (1997) analysis of his family's grocery list. We'll limit the scope of the analysis to one weekly shopping trip cycle that includes preparation for the shopping trip.

Russell and his daughter generally do the grocery shopping so we shall designate them as the *subjects*. The *object* is the list of items to be purchased at the local grocery store. The family creates this list over the course of the week. The desired *outcome* for the activity is for the family to have the supplies they want and need once the shopping trip is complete. The family (*community*) keeps a photocopied list of items (*tool*) they regularly buy posted on the family refrigerator (*rules, norms, conventions).* Each family member (*community*) marks items on the list they want or need (*division of labor*). One might imagine family members add items to the list as they go about their routine activities. Perhaps the daughter adds candy to the list because her class Valentine party is in the coming week. Another member of the household removes the last roll of toilet paper from the cabinet and notes that the family needs toilet paper in the next shopping trip. Someone else in the family might want a special dessert and add the necessary ingredients to the list.

2.2 Networks of Activity

The CHAT framework recognizes the networked nature of activity.

- Within a central activity, other activities are embedded or nested. Engeström (1987) calls these object-activities because they all share the same object and outcomes.
- There are also nearby activities focusing on the production of tools to be used in the central activity.
- Subject-producing activities focus on recruiting, training and educating subjects or potential subjects for the system.
- Finally, rule-producing activities focus on creating rules, policies, and/or legislation impacting the central activity. Frequently, activities occur in parallel.

Let's return to Russell's (1997) grocery list example. The object-activities are all of the activities that contribute to making the grocery list. In the case of the family member wanting to make a favorite dessert in the coming week, retrieving the recipe for the dessert, noting the required ingredients, checking the kitchen cabinets to see what is on hand, and marking the list for required items all contribute to making the grocery list for the next shopping trip and are object-activities. Russell and his daughter have recruited other family members (subject-producing) to help make the list. He and his daughter have organized the list (tool-producing activity) to correspond with the aisles in the grocery store – an influence from the nearby activity of the actual shopping trip.

Because of the networked nature of activity, various elements of the activity can move from one node in the framework to another throughout the course of the activity. When Russell and his daughter (the subjects in his example) go to the grocery store, the role of the list shifts. While they were creating the list, it was the object, but at the store it serves as a tool, to help them complete their shopping trip (the new object). For many shoppers, the list (first the object, now a tool) has no further value and is discarded on the way out of the grocery store. However, in the Russell family, the list is retained because it is further used as a tool to mediate family arguments about what was purchased or not purchased by the item's presence or absence on the list. One of the norms in the Russell family is that if a family member has forgotten to put an item on the list, he or she has no basis for complaint if and when the desired item is not purchased. Although there is not a specific rule-

producing activity (signing a contract or passing a new law), over time, the family eventually developed the norm that the shopping list would be used to mediate disputes about purchases.

2.3 Tensions

Engeström's (1987) theory further describes "tensions" or "contradictions" that can arise within an activity.

Software developers study this concept under a different name, the "forces" of design patterns:

> "A pattern describes a solution to conflicting forces, to quote Christopher Alexander, in such a way that you can use this solution a million times over, without ever doing it the same way twice." (PortlandPatternRepository, undated)
>
> "A pattern is more than just a battle-proven solution to a recurring problem. The problem occurs within a certain context, and in the presence of numerous competing concerns. The proposed solution involves some kind of structure which balances these concerns, or "forces", in the manner most appropriate for the given context." (Appleton, 2000)

An advantage of activity theory is that it provides an explicit structure for thinking about different types of tensions, so that we can more easily imagine the diversity of competing concerns.

In activity theory, tensions develop within and between nodes of the activity. Engeström (1987) organizes these tensions into four distinct layers.

- The first, or primary, level of tension occurs within an element of the activity.
- Secondary tensions can arise between two different nodes of the activity.
- A tertiary tension arises between one form of activity and a culturally more advanced form of the same activity.
- The final layer of tension, quaternary, is a tension between nearby activities.

Engeström (1987) argues that tensions can motivate innovations as subjects change their activity to reduce or eliminate these tensions.

The classic discussion of tensions within an activity comes from a retrospective analysis of the Challenger shuttle accident by Holt and Morris (1993). Their work is a good tutorial on analyzing the first three layers of tension. Their analysis looks specifically at the Flight Readiness Review (FRR) in place at the time of the accident. The tensions stemmed from the competing pressures to put safety first and to make 24 flights per year.

- In the rules node, Holt and Morris (1993) identified "safety first vs. timely flight" as a primary tension. The decision makers (subjects node) also faced a primary tension as they tried to put safety first while at the same time being cost conscious. The FRR tool had a primary tension as decision makers used "the FRR as a checklist that NASA follows 'to the letter' (what one might call the 'hard reading'), as opposed to an FRR that NASA follows only when convenient" (p. 105).
- One of the important secondary tensions in this analysis is the one between the decision makers and the FRR tool as decision makers tried to juggle the competing pressures of safety and cost. The tragedy of the Challenger accident moved safety to a higher priority within the FRR.
- According to Holt and Morris' (1993) analysis, at the time of their analysis, the system was at a tertiary tension because the system had changed to make safety a higher priority even though the cost pressures remained. The system had not yet resolved the pressures of the "old" way of making decisions with the "new" way.

Tensions are important to the activity theorist because they give rise to "need states." Actors in the network seek ways to satisfy needs the existing activity system does not satisfy, resulting in change.

More generally, thinking about a system in terms of its tensions grows out of Hegel's conception of the dialectic (Engeström, 1987; Giest, 2008; Hoffmann, 2005; Holzman, 2006; Seaman, 2008). By discovering the conflicts between two concepts, we gain a deeper understanding of each.

- To the extent that we are trying to build an accurate model of something, we gain insight into the model by looking for the contradictions between its predictions or descriptions and those of other credible models. Probably, that insight leads us to improve the model.
- Similarly, if we are trying to create or improve a product or process, we learn a lot about what we are proposing by thinking about the ways that the new/improved version can cause problems. It's not just about finding bugs in the

proposal. It's about gaining a deeper understanding of the proposal and its context and constraints so that the next version is built on a better foundation.

As testers, we understand the value of evaluating a proposal by thinking about the impact on each stakeholder, the interoperability with each device, client or server in the system, the errors made more likely, the performance made slower, and the tasks made more confusing or less convenient. We focus on these issues with scenario-based and risk-based testing. When we file bug reports, we expect the developers or designers who are creating the product to not just fix the specific bug reported but to also assess the proposed system and imagine, when appropriate, a stronger design that eliminates the underlying weakness at the root of the specific misbehaviors reported. Activity theory trains the person who uses it (designer, programmer, tester, business analyst, marketer, etc.) to imagine a broader collection of people or systems that might be influenced and the several ways (positive and negative) in which each one might be affected.

2.4 Changes in an Activity System

Changes in an activity system can come from within the system or from nearby systems. The changes occur as the subjects identify new or better ways to resolve tensions or satisfy needs not satisfied by the existing system. Such changes or innovations might remain with an individual, changing only that individual's activity. However, these innovations are frequently shared with other actors and, where appropriate, are adopted by others. An activity system is changed these innovations.

The Russell family's grocery list included an innovative change when Russell and his daughter decided to organize the list that is kept on the family refrigerator to correspond with the aisles in their favorite grocery store. Thus, Russell and his daughter resolved a quaternary tension between list-making (one activity) and shopping (a nearby activity) by introducing their newly-organized grocery checklist (*tool*) into the list-making activity as a result of the shopping trip activity.

Engeström (1987) offers an interesting example from the history of Chemistry. In essence, chemists in the latter half of the 19th century had a well-entrenched system of organizing the elements by similarities. At that time, organizing the elements by atomic weight (as we do today) was unthinkable because dissimilar elements would be placed next to each other. For D.I. Mendeleev, the need to organize the elements became urgent as he progressed with his textbook *Fundamentals of Chemistry*. According to the account Engeström (1987) reports, Mendeleev had written two chapters of his book and had to decide which chapter to include next. He was under time pressure because of upcoming travel when he received inspiration for organizing the elements by property and atomic weight while playing solitaire. Recall that the table is organized vertically by property and horizontally by atomic weight. Thus, the modern Periodic Table of Elements was conceived during a game of solitaire.

3. CASE STUDY OF *COMMERCIALFOLIO* AT VENDORBUILT COLLEGE

The Russell family example provides a simple context for introducing CHAT's key concepts. This second example illustrates the use of CHAT at a real-world level of complexity.

CommercialFolio is a fictitious name for a real software product that Fiedler (2006) studied in depth at VendorBuilt College (fictitious name for a real college). Several different types of users work with CommercialFolio or its results. These include students, faculty, administrators (department chairs, college deans, etc.), and accreditors (groups of people who evaluate the quality of the college and its fitness to offer the degrees that it offers). CommercialFolio collected and organized data that helped administrators demonstrate the quality of student work to accreditors. There was also a belief (especially among vendors of CommercialFolio and some college administrators) that the CommercialFolio data would help faculty appraise student work and help students get jobs because the CommercialFolio data would provide potential employers with an extensive sample of the student's work. The job-seeking benefit was seen (by the vendor and administrators) as so valuable to students that the university could require students to pay the subscription fee for the privilege of working with CommercialFolio. They believed that students would appreciate this benefit so much that, after graduating from VendorBuilt, they would continue to pay for ongoing access to their own data stored in CommercialFolio's database.

This case study illustrates using qualitative research methods to gather data about the student experiences with CommercialFolio and CHAT to guide the design and progress of the research and the interpretation of the results.

As is so often the case when the same software serves many different groups of stakeholders, a powerful stakeholder whose needs are met by a product might believe that a less powerful stakeholder with different needs and interests will love the product too, but that doesn't necessarily make it so. In VendorBuilt's case, the students were not delighted with being required to use (let alone pay for) this software. Among the many issues that surfaced in the research, students didn't have enough control over what was (or was not) stored in CommercialFolio, or how attractively their data would be formatted, to want to use it as a primary source of disclosure to prospective employers. Given the student reactions that surfaced in this study, it seems overly optimistic to expect a significant

revenue stream from students who voluntarily pay for the opportunity to keep using this software after they graduate.

Fiedler worked mainly with students, who were preservice teachers (students studying to become teachers). She did not research the development of CommercialFolio. She sometimes spoke with sales staff or technical support staff, but not designers, programmers, human factors analysts, or software testers. We do not know how well CommercialFolio's development staff anticipated the reactions of student users. What we do know is that if CommercialFolio's testing staff did traditional functional verification as their "user acceptance testing" or limited their tests to user stories or scenarios developed by accreditors, university administrators, and internal-to-CommercialFolio developers, those testers would have been oblivious to the problems that CommercialFolio would pose for many students and the perceived inadequacy of the tool as a support for job search.

If your role as a tester includes determining whether users will like the product (or whether there are aspects of the product that will make some users hate it), you will need to identify different types of users and study how each one will interact with the product. As the following case study will illustrate, CHAT provides a support structure for this type of analysis and research.

Lee Shulman (1998) defines a teaching portfolio as "the structured, documentary history of a set of coached or mentored acts of teaching, substantiated by samples of student portfolios, and fully realized only through reflective writing, deliberation, and conversation" (p. 37). For preservice teachers, a portfolio is a collection of lesson plans and class activities they've created. Each entry in the portfolio is accompanied by a note explaining why they did what they did, how it worked, and further improvements they would make based on what they learned from using it. Taken together, the portfolio entries and accompanying discussions provide evidence the preservice teacher is ready to enter the classroom as a fully credentialed teacher.

In recent years, electronic portfolios (digital collections of work) have become a hot discussion topic on colleges and universities across the country. Batson (2002) has written about "the electronic portfolio boom" and Cohn and Hibbits (2004) have called electronic portfolios "higher education's new 'got to have it' tool" (p. 1). The convergence of widespread Internet access and web-enabled databases has contributed to the emergence of a new breed of electronic portfolio that allow administrators to collect and aggregate data for a variety of purposes: documenting progress toward meeting accreditation standards, program evaluation, and student achievement. Electronic portfolios are a widespread phenomenon on college and university campuses – extending beyond teacher education programs. Electronic portfolio initiatives are generally high profile and high-stakes initiatives.

CommercialFolio[1] is one of the large commercial providers of this type of service. They offer a Web-based tool designed to help high school and college students compile electronic portfolios of their work to satisfy demands of various accrediting agencies. *CommercialFolio* markets their product to faculty and administrators in charge of institution's accrediting efforts. In their marketing, *CommercialFolio* stresses the ease and convenience of data collection and reporting for accreditation purposes.

Once decision makers decide to adopt and implement the *CommercialFolio* solution, they require each student to purchase the *CommercialFolio* subscription and use the subscription to compile a portfolio as a condition for graduation. There is no direct cost to the adopting college or department. The student's initial subscription lasts for three years and gives students access to the service throughout their program of study and for one year beyond. After the initial subscription lapses, students may purchase a renewal.

This case study is set within an education department, but similar initiatives are happening across campuses throughout the country.

VendorBuilt College (VBC) is a small liberal-arts college in the southeast. Students in the education department at VBC are encouraged to start their portfolios early and one of the faculty members developed a template to assist students in this task. Figure 2 illustrates a page from the VBC template. Most student portfolios look much like this when they are completed.

[1] Throughout this paper, we refer to individuals, institutions, and vendors by pseudonyms. Thus, *CommercialFolio* is not the company or product's true name.

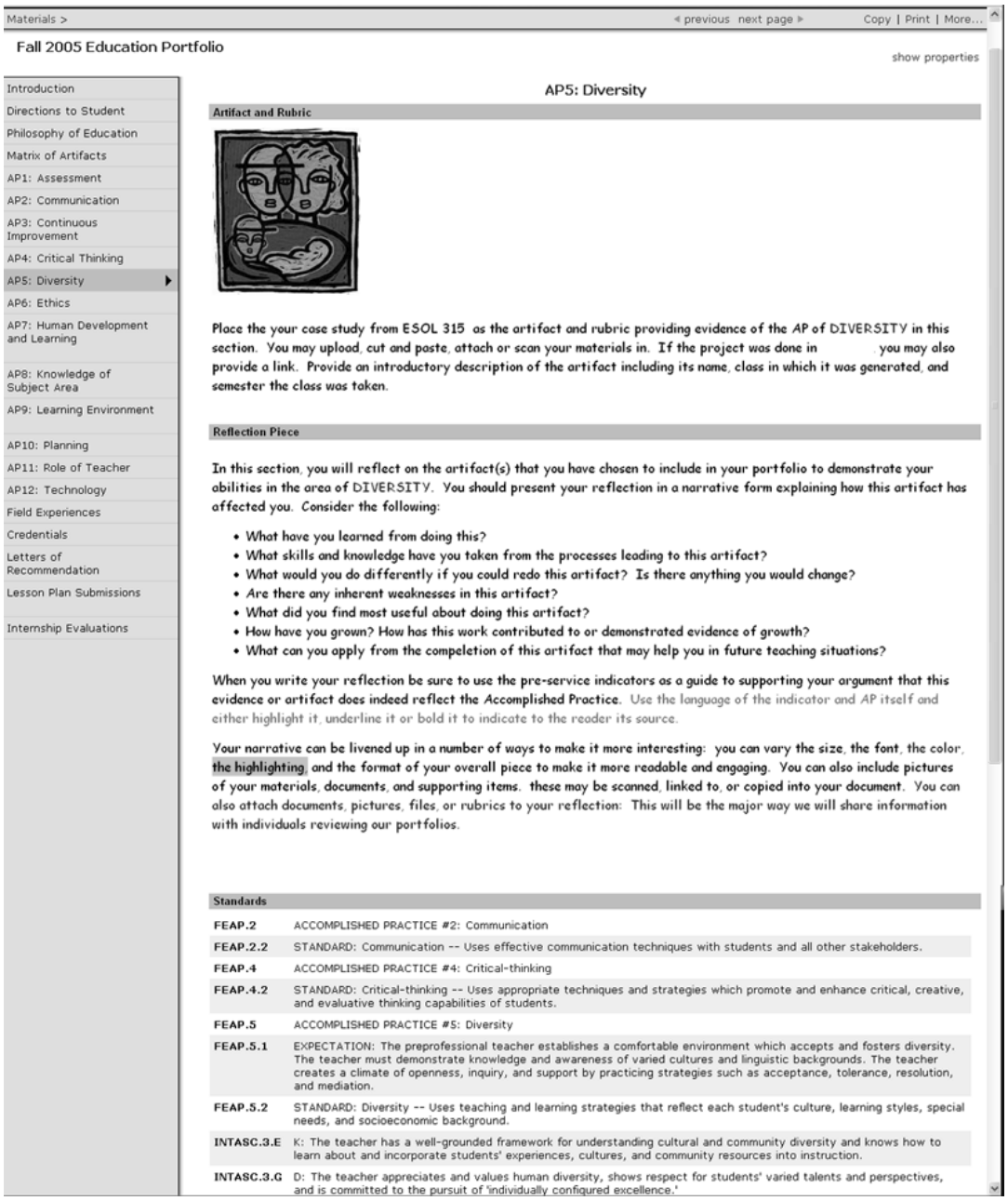

Figure 2: Page from the VBC portfolio template includes three sections separated by colored bars.

The template and all portfolios are organized around the state's 12 standards for teachers. These standards outline generic skills all teachers should have in the areas such as communication, assessment, knowledge of subject matter, planning, and others. For each standard, students must provide one or more artifacts from their work to demonstrate their mastery of that standard along with the rubric (grading notes from faculty). In addition to the artifacts and rubrics, students must also write a reflection on the artifact outlining why the included artifact offers compelling evidence of mastery of the standard along with a narrative discussing how the artifact helped them to develop skills, how the student might change the artifact, any inherent weaknesses, and/or how they might apply what they've learned to their future careers. Portfolio visitors navigate the portfolio using the links on the left of the portfolio.

The navigation bar on the left side allows students and portfolio reviewers to navigate through different sections of the portfolio. The tab labeled Matrix of Artifacts includes the list of specific requirements. This matrix is also distributed to the students as an Excel file.

To use the *CommercialFolio* service, students must work at an Internet-connected computer with a Web browser. Each subscriber/student receives a user name and password to log on to the *CommercialFolio* system. Students can upload documents as attachments or enter text into a text entry field. The system allows images to be uploaded and displayed on portfolio pages. Each page has a Save button to save work that is in progress.

Students can generate visitor passes to allow others to see their work. At VBC, it has become common practice for seniors to share their portfolios with younger friends who are looking for ideas about what to do with their own portfolios. Subscribers can also share their work with other *CommercialFolio* subscribers as editors or reviewers. Professors often require students to add them as reviewers so they can make comments on specific pages. Portfolio authors can review the comments as they make modifications. Professors can also develop rubrics specific to assignments and grade the students according to those rubrics. Some professors use this electronic capability while others prefer to do grading in paper format.

Students shared the following experiences and anecdotes[2] in a series of interviews and thinkaloud work sessions. We will return to analyze them later in this paper.

> CATHY: I have Dr. Lancaster for two classes and she assigns these big, huge projects and both of these projects are required to be included in my portfolio and she knows it. But she won't let us turn our projects in electronically. She makes us print them out and everybody knows she likes all of this creative stuff and we have to do all this "cutesy" stuff to get a good grade. Then, I have to do just as much work all over again just to reformat everything to go into the portfolio. I don't have time for this. Most of the time the scanners in the lab aren't working and so it's really hard to scan the things that need to be scanned. I did a lot of the project using *Microsoft Word*, but when I try to copy it from *Word* and paste it into the tool, everything just goes crazy. I end up with all of these spaces that I didn't put there and sometimes it takes away some of my formatting and it adds other formatting and I spend hours just trying to clean that up and it just doesn't work. This was a 36-page paper and who has time to retype everything? I know I don't.
>
> PHOEBE: We spent a lot of time talking in class – those of us in the same major. We were allowed to pick four artifacts, but we didn't feel like we had good ones for any of those four standards. So we sat around trying to figure out how we could turn a paper on the Civil War into a communication artifact. It was just ridiculous, but whenever somebody had a good idea about one thing, we'd all kind of try to do the same thing.
>
> HANNAH: Look! Look! I have two pictures in one section and I didn't think *CommercialFolio* could do that. How did I do that? I don't know how I did that! Everybody, come see this!
>
> ANNE: I was just *so* stressed out about all of this. You see, I have to have this done because it's a graduation requirement and I felt like I was behind and had to get caught up. The service was running really slow and I didn't get everything done I needed to get done before the computer labs closed the other night. So, since I don't have Internet access at home, I went home to get my roommate's laptop. She has a wireless card and the library has wireless access. So I came back in the middle of the night and I went and sat on the library steps after it closed so I could pick up the signal and work on my portfolio. It was really creepy because this scary-looking guy came by. I kind of moved so he wouldn't see me and, lucky for me, he didn't. After he was gone, I stayed a lot closer to my hiding place, but I kept working. I had to. I got everything done, too.
>
> ALICIA: I've learned a lot about HTML from using *MySpace*. I don't know if you're familiar with it or not, but I think it's probably taught a lot of our generation how to do HTML. I can put on backgrounds and pictures and make links in my profile. I spend a lot of time using *MySpace*.
>
> TUCKER: My professor graded my project and filled out the rubric or grading sheet using this portfolio tool. I'm required to show that in my portfolio and I don't know how. The professor said she doesn't know how either except for me to print it out, scan it, and then add it as an image. That doesn't make sense to me. I want to see if I can figure out a better way because I have to do this – like 30 times – and I don't have a scanner at home. Between my classes and my part-time job and trying to have a life, I don't have time to come to the computer lab hoping that (a) the scanners are working and (b) that I can actually use one when I need it.
>
> RAFAEL: You need to understand the time that's involved in it.

[2] The *CommercialFolio* product and student quotes are based on data from Fiedler's dissertation. The quotes have been synthesized and are not from any specific person but they do reflect actual concerns of real students using VBC's portfolio authoring tool.

3.1 Fleshing Out the Model

The first step in a CHAT analysis is to saturate the model. Recall that the *subject* may be an individual or a group and that the choice determines the perspective or point of view for the analysis. In this case study, Fiedler (2006) chose to designate the preservice teachers as her subjects. These young men and women were completing their undergraduate degrees in preparation for careers in teaching. For the case study, she might also have chosen college faculty, administrators, accreditors, or potential hiring authorities as the subjects for analysis.

The object carries the motive or purpose for an activity. It's important to note that objects and purpose may evolve (Engeström & Blackler, 2005) and the portfolios at VendorBuilt College were often in flux. In this analysis, the selected object was the electronic portfolio students were creating. Students had a variety of motives or purposes for the portfolio task. From the student perspective, the fact that creating the portfolio is a requirement for graduation was certainly a motive for working on it. For many students, the graduation requirement was the only motive but other motives included record-keeping for the learning and the work they've done; usefulness for job hunting; and plans for advanced degrees or credentials such as the prestigious National Board Certification.

In the CHAT model, *tools* mediate the interaction between subject and object and can be tangible, symbolic, or psychological. The students in the case study used a wide range of tools including the *CommercialFolio* subscription system for portfolio authoring; faculty-created handouts and templates; computers; scanners; course syllabi; electronic help resources; tech support; word processing and presentation software; paper; pencils; colored markers; previous work samples; and many other things. Students even use their professors as tools when they asked for advice or turned to them for explanations.

The *community* is the group of individuals who share the same object. In the case of students creating electronic portfolios, the community is comprised of other students, the faculty (yes, they are in more than one area of the model), potential hiring authorities, accreditors, and friends and family of the students involved in the portfolio task.

The intersection of community and object is labeled *division of labor*. In the case study, labor was divided both horizontally and vertically. Horizontal division of labor was evident in the portfolio system Fiedler (2006) examined as students helped each other think about some of the decisions that went into the portfolios, peer reviewed each others work, and helped each other complete various tasks. Vertical division of labor refers to tasks divided across the different divisions of power. In this case, faculty developed templates and handouts to help students with their portfolio task.

Finally, the interaction of the subject with the community is typically governed by *rules, norms, and conventions* whether formal, informal, or technical. In the *CommercialFolio* case study, rules included the requirement to complete a portfolio prior to graduation; the requirement to use the *CommercialFolio* subscription software; and rules governing format and structure of the portfolio. Many of the rules regarding format and structure came from the outside accrediting agency. Norms included the custom of senior students sharing portfolios for their younger peers to review as well as the students' practice of getting together to "work on" the portfolios.

3.2 Networks of Activity

In a CHAT analysis, the analyst looks for other activities in the network. Those activities may be embedded within the activity of interest or nearby. In the *CommercialFolio* case study, a *theoretically* interesting component of the portfolio activity system was a document called the "Matrix of Artifacts." Tracing the matrix of artifacts as it moves through the portfolio activity system provides an illustration of the networked nature of the system surrounding the *CommercialFolio* software in use. To follow the matrix through the activity system, one must examine nearby activities requiring temporary adoption of perspectives other than the students'.

The matrix of artifacts is reproduced in Table 1 to illustrate the basic attributes of this relatively simple document. The matrix lists courses and required artifacts in the two left columns. The labels across the top represent accrediting standards and the Xs indicate which standards a particular artifact satisfied. Working with the matrix surfaced serious defects in *CommercialFolio's* table-making capability.

Table 1. Matrix of Artifacts

Courses	Artifacts	AP 1	AP 2	AP 3	AP 4	AP 5	AP 6	AP 7	AP 8	AP 9	AP 10	AP 11	AP 12
EDU 406	Personal Orientation			X	X							X	
(Curriculum)	to Curriculum Paper												
EDU 357	Behavior Management		X							X			
(Behavior Management)	Plan												
EDU 203	Test Construction	X									X		
(Tests/Measurements)	Project												
EDU 324	Multimedia		X										X
(Instructional Design)	Project												
EDU 321	Lesson Plan	X							X		X		
(General Methods)													
EDU 321	Unit Plan								X		X		
(General Methods)													
ESL 315	Case Study		X		X	X							
(Cross Cultural Comm.)	Ethnography												
ESL 325	Applied Linguistics					X			X			X	
(Applied Linguistics)	Project												
ESE 209	Connections Project												
Introduction to Exceptionality	New and Scholarly Literature							X	X				
EDU 481	See Student Summative												
(Internship)	Forms Folder												
EEC 209	Case Study:							X	X				
(Intro to Exceptionality)													

Let's begin with the creation of the matrix. It was designed to communicate requirements to students, so we need to go back a bit in time for faculty and administrators to outline what they planned to require of student portfolios (a rule producing activity). These rules were reduced to writing and displayed in the matrix to duplicate and distribute to students (a tool-producing activity). Once students had the matrix, many used it to keep track of their requirements. They used it as a tool to monitor their own progress, marking it up in a variety of ways. In addition, students were required to update the matrix (an object-producing activity) and include it in their portfolio to help a portfolio reviewer understand where different artifacts were electronically filed (tool for reviewer). Many students chose to re-organize the matrix and decorate it with color before putting it back in their portfolio. As they performed these updates, the matrix became the object of their attention[3].

4. Tensions within the *CommercialFolio* Case Study

To us, the most interesting use for CHAT is to analyze tensions within an activity. As noted in Section 2.3, tensions are inevitable in a complex system. Tensions develop within and between nodes of the activity resulting in four possible layers (Engeström, 1987). Study of tensions can yield new insights, changes, and innovations as participants in an activity look for ways to satisfy unmet needs and resolve tensions. We examine the idea of tensions and changes in software contexts by returning to the portfolio case study. Although tensions are sometimes productive, in this report for software testers, we are

[3] The CHAT model is fluid. Anyone seeking to use CHAT to analyze a situation will notice that the nodes on the model tend to be "slippery" as the attention shifts from one part of the model to another or moves from big picture to small picture views. For example, the matrix of artifacts is a tool for the portfolio activity. Before that, it was an object as faculty created it. It became an object again as students updated the matrix. Once completed, the matrix returns to the role of a tool for portfolio reviewers to use.

focusing on tensions that *CommercialFolio* developers might have avoided or minimized by a more thorough understanding of prospective users.

Primary tensions are conflicts that reside within a specific node of the framework. The matrix of artifacts document offers examples of primary tension within the activity system. With *CommercialFolio*, students used the Table creation tool in Edit mode to reproduce their Matrix of Artifacts. When they switched to View mode, they discovered that *CommercialFolio* did not properly render the table. Repeated attempts to edit the table were unsuccessful because *CommercialFolio* rendered the tables inconsistently. Within *CommercialFolio's* table tool, table creation and table rendering didn't work.

In addition, students tried to copy their tables from Microsoft *Excel* documents and paste them into *CommercialFolio* but were similarly unsuccessful and unable to edit the results to fix the table display. For students, the incompatibility between a major productivity tool and *CommercialFolio* wasn't confined to table functionality, either. Cathy reports on some of the challenges she faced moving class projects into her electronic portfolio.

> I did a lot of the project using Microsoft *Word*, but when I try to copy it from *Word* and paste it into *CommercialFolio*, everything just goes crazy. I end up with all of these spaces that I didn't put there and sometimes it takes away some of my formatting and it adds other formatting and I spend hours just trying to clean that up and it just doesn't work.

From the students' perspective, tools they ordinarily used were incompatible with *CommercialFolio* and they were frustrated by that fact.

Given the purpose of portfolios in today's higher education context, *CommercialFolio* should have expected that students would move already completed products into their system. If they didn't realize this by thinking through how people would use the system, we think that this type of problem would have shown up quickly in user-oriented testing. This is not a subtle problem. To find a problem like this does not require carefully controlled experiments, fancy modeling, or sophisticated statistics. This is not the kind of problem that only a human factors specialist could find--unless only human factors specialists are allowed to look for problems like this. Moderately skilled testers can find problems like this--but only if they look for them.

Secondary tensions are those tensions that exist between any two nodes of an activity system. In this system, *CommercialFolio's* capacity to display tables was at the heart of a secondary tension between the *rules* and *tools* nodes. Students were required to include their matrix of artifacts in their portfolio and they were required to use the *CommercialFolio* tool even though it did not properly render the required tables. Students wasted enormous amounts of time trying to resolve this "double bind" situation. Some repeatedly tried to edit their tables with the hope that the table would somehow display properly. Several tried to create or modify the tables by working in the HTML code. Others used *CommercialFolio's* file attachment feature to add a *Word* or *Excel* file although they deemed this an unsatisfactory solution because the Matrix was not clearly visible to portfolio reviewers. Many students simply abandoned their efforts to fix the tables and left the matrix in their portfolios in whatever form it happened to appear.

At VBC, students repeatedly confronted secondary tension between the *CommercialFolio* tool and their requirement to use it. In a more flexible system, students might have found a workaround solution but the requirements at VBC made that difficult. Anne explains how she found it difficult to meet deadlines because of *CommercialFolio's* performance and what she did about it in this excerpt from the research transcripts.

> I was just so stressed out about all of this. You see, I have to have this done because it's a graduation requirement and I felt like I was behind and had to get caught up. The service was running really slow and I didn't get everything done I needed to get done before the computer labs closed the other night. So, since I don't have Internet access at home, I went home to get my roommate's laptop. She has a wireless card and the library has wireless access. So I came back in the middle of the night and I went and sat on the library steps after it closed so I could pick up the signal and work on my portfolio. It was really creepy because this scary-looking guy came by. I kind of moved so he wouldn't see me and, lucky for me, he didn't. After he was gone, I stayed a lot closer to my hiding place, but I kept working. I had to. I got everything done, too.

Recall that a tertiary tension is a conflict between one activity system and a culturally more advanced form of the same activity system. In the case study, Cathy told us about one of her professors:

> I have Dr. Lancaster for two classes and she assigns these big, huge projects and both of these projects are required to be included in my portfolio and she knows it. But she won't let us turn our projects in electronically. She makes us print them out and everybody knows she likes all of this creative stuff and we

> have to do all this "cutesy" stuff to get a good grade. Then, I have to do just as much work all over again just to reformat everything to go into the portfolio. I don't have time for this.

Obviously, this is a tension between the professor's traditional view of paper-based work products and the contemporary requirement for students to create electronic portfolios.

In addition, students were required to include a grading rubric[4] with the corresponding assignment in their portfolios. In many cases, the students only had paper copies of rubrics from earlier classes and had to find a way to add it to their electronic portfolios. Recreating the rubric using *CommercialFolio's* table tool was not a viable option so many chose to scan their paper rubrics, resize the resulting files, and add the scanned documents as images. Students widely perceived the scanning solution as burdensome because many didn't have scanners of their own and scanners in the computer labs were often already in use or not working.

In addition to rubrics, students also had to scan evaluations they received on paper from teachers in local schools, observations of their student teaching, student observations in local schools and artifacts they wanted to include in their portfolios. The sheer volume of scanning required to satisfy portfolio requirements was a daunting task. Rafael explained, "You need to understand the time that's involved in it." He pointed out that problems crop up and little things added up to a lot of time: time to do the artifact, go match it to an accomplished practice in the words of the accomplished practice, write the reflection, do the revisions. There were 12 artifacts and all of the revising and scanning that go along with each of them. At our interview, he was days away from graduation and had spent seven hours the previous day trying to scan documents and had been on campus for another three hours on the day I spoke to him. His chief complaint at this point was that there was a disproportionate amount of stuff to be scanned: many pages of evaluations, 12 practica, six observations, 12 rubrics, and a picture on the splash page. He was quick to point out that scanning images also required resizing them as well.

This tension is between the old way – paper documents – and the new requirement to put the documents in the electronic portfolio.

The final type of tension is quaternary or a tension between the central activity and nearby activities. In Fiedler's interviews with students using *CommercialFolio*, students discussed the various features and shortcomings of *CommercialFolio* by referencing their experience with other tools they had used including the popular *MySpace* social networking site and standard productivity software. Their familiarity with procedures and capabilities in the other software and services played a role in setting their expectations for *CommercialFolio's* capabilities. To the extent that *CommercialFolio* exceeded their expectations, they were delighted. To the extent that *CommercialFolio* fell short of their expectations, they were disappointed.

It's not unusual for different kinds of tensions to be entangled to make CHAT analysis challenging. To illustrate this, we consider one of the things Tucker discussed in an interview.

> My professor graded my project and filled out the rubric or grading sheet using this portfolio tool. I'm required to show that in my portfolio and I don't know how. The professor said she doesn't know how either except for me to print it out, scan it, and then add it as an image. That doesn't make sense to me. I want to see if I can figure out a better way because I have to do this – like 30 times – and I don't have a scanner at home. Between my classes and my part-time job and trying to have a life, I don't have time to come to the computer lab hoping that (a) the scanners are working and (b) that I can actually use one when I need it.

In some classes, professors used *CommercialFolio* to grade student work using a rubric. *CommercialFolio* appeared not to offer the capability of showing a rubric created on the instructor side of the system alongside the work that had been graded. At first look, this appears to be a primary tension within the Tools node. An alternative interpretation is to class this as a secondary tension between the Rules and Tools node because students were required to include those rubrics with their work but *CommercialFolio* didn't appear to offer that functionality. When students and faculty called the company to find out they might resolve this, representatives said they didn't provide that functionality because it was inappropriate to make grading information visible to portfolio visitors. VBC faculty and accreditors nevertheless required it and so one might correctly categorize this tension as quaternary or between nearby activities – between portfolio creation and that of the software company or the accrediting body. Students in Tucker's situation – and there were many – then confronted the choice of trying to recreate the rubric using the poorly-functioning table tool in *CommercialFolio*; creating the table in their word processor and hoping for the best as they pasted it into *CommercialFolio*; or printing the rubric, scanning it, and adding it as an image

[4] A grading rubric is often in the form of a table. Most rubrics list the criteria for a specific assignment and delineate several levels of performance for each criterion. The rubric is used to determine a grade for an assignment. For examples, see *RubiStar* (4Teachers.org, undated).

as Tucker's professor suggested. For all students, this portfolio activity used precious time they needed for other classes, part-time jobs, and time with family and friends. In CHAT terms, the portfolio activity competed with nearby activities that were also important to the students.

The problems that students had with tables illustrate another example of a cluster of issues that should have been obvious and easily discovered by a tester who assessed the product from the student-user's point of view. CHAT would help the tester identify the different types of users, the types of tasks they are likely to do, and to recognize the importance of some of the problems that a user runs into. But it is not the decision to use CHAT that is critical. CHAT is just a very useful tool for helping testers adopt a system-level viewpoint and applying it to evaluation of the product's quality. The critical decision behind this is the decision to break out of the tight little box of features, functions, and a few predefined "stories" and to operate in the worlds in which the product will be used.

The problems that exasperated these students were not the kind that would have been exposed by a tester who studied a little more programming and wrote a suite of unit tests. Black box system testing isn't about the absence of knowledge of the code. It's about the application of knowledge of the product's context. That requires study of the ways in which the product will be used, the people who will use it (and their expectations, needs, and preferences), the devices and software it has to interoperate with, the conflicting and often-incompatible demands that have been made on the product's design, and the nature of the consequences that the various product failures can cause.

4.1 Changes in the *CommercialFolio* Activity System

Changes in an activity system can come from within the system or from nearby systems. Some are relatively simple to implement. Phoebe talks about how she and other students in the same major worked together and shared their best ideas with each other.

> We spent a lot of time talking in class – those of us in the same major. We were allowed to pick four artifacts, but we didn't feel like we had good ones for any of those four standards. So we sat around trying to figure out how we could turn a paper on the Civil War into a communication artifact. It was just ridiculous, but whenever somebody had a good idea about one thing, we'd all kind of try to do the same thing.

Other changes are more complex. The *CommercialFolio* case study had an accidental discovery that eventually led to long-lasting change to the system. But first, some background.

CommercialFolio generates web pages with sections. As designed, each section can have one and only one graphic or image. Placement options are limited to left alignment, right alignment, or center placement. To add more images, users must add more sections and those sections are separated by a colored bar. See Figure 2 for an example. Portfolio authors Fiedler interviewed viewed this restriction as a serious limitation of the *CommercialFolio* tool and they resented this constraint on what they could do. In fact, *CommercialFolio's* image handling was a frequent complaint by the users Fiedler interviewed. They compared *CommercialFolio* unfavorably to other products with which they were familiar. In CHAT terms, their participation in the nearby activities of using word processors and *MySpace* set their expectations for *CommercialFolio's* capabilities. They experienced quaternary tensions when *CommercialFolio* failed to meet those expectations.

Anne was one of the most technically skilled users among the preservice teachers at VendorBuilt College. She credited much of her skill to what she learned using *MySpace*.

> I've learned a lot about HTML from using *MySpace*. I don't know if you're familiar with it or not, but I think it's probably taught a lot of our generation how to do HTML. I can put on backgrounds and pictures and make links in my profile. I spend a lot of time using *MySpace*.

Anne was determined to find a way around *CommercialFolio's* image constraints. She decided to try something she had learned from her experience with *MySpace* in her *CommercialFolio* portfolio. She worked in HTML view to use <img_src> tags and a free photo hosting service to resolve the limitation of one photo per section. However, Anne was reluctant to share her solution because she was worried it was "not allowed" and that *CommercialFolio* would somehow "break" what she did. Read details of Anne's experience in (Fiedler, 2007).

Hannah was the first to put two images in the same section of her portfolio publicly. It happened during class and Hannah squealed with delight, insisting her professor and classmates come see it for themselves.

> Look! Look! I have two pictures in one section and I didn't think *CommercialFolio* could do that. How did I do that? I don't know how I did that! Everybody, come see this!

Despite several attempts, she could not replicate her actions. A couple of weeks later, Alicia made a similar accidental discovery but was able to replicate what she did. From there, Alicia used her knowledge of *MySpace* in conjunction with a

free photo hosting service to leverage her discovery into a way for her to post as many images as she wanted without worrying about *CommercialFolio's* limitations. Alicia independently arrived at the same solution Anne had.

By the end of Fiedler's first research visit, Alicia and Anne shared their discoveries with their professor and some of their peers. Other students found it very interesting but did not widely adopt it because they shared Anne's suspicion of the durability of the solution. Two years later, Fiedler returned to the same institution for more research and saw that Alicia and Anne's solution workaround to the image constraints was in wide use. Further investigation revealed that *CommercialFolio* representatives had discovered many students across a variety of institutions were using the same workaround. *CommercialFolio* staff made one final revision to the process and incorporated it into their training of faculty who, in turn, routinely passed it along to students.

5. OUTLINING SOME APPLICATIONS TO SOFTWARE TESTING

Now that we've looked at CHAT as a general approach to studying systems, and changes within systems, let's sketch a few possible applications of CHAT. To maximize your benefit from this paper, we suggest that you work with a small group and work through a concrete example (such as a specific new tool). These sketches are designed to help you structure that discussion, but it is the insights you gain as you consider new tensions, and then shift viewpoints and discover yet more tensions, that drives home the analysis. We cannot do that for you in a paper. You have to have your own "AHA!" moments.

Consider these situations:

- Introducing a new metric to assess staff productivity or product quality or something else;
- Introducing a new test tool to the test organization, not all of whom have or want to develop the technical skills expected by the designers of the tool;
- Interviewing stakeholders to gather their requirements (needs and preferences) and to discover the conflicts among stakeholders' requirements (good scenario testing rests on insight into stakeholder requirements);
- Describing failures (in bug reports and triage meetings, for example) in ways that are intended to motivate specific stakeholders to demand fixes.

5.1 The New Metric

You lead a test group. As your company considers ways to cut costs, each group is asked to justify its existence and its staffing level. Famous Consultant has advised your executives to use Metric P to evaluate the productivity of your staff, and your executives have asked you to compute Metric P on a regular basis and report its values to your staff and to them.

We are intentionally ***not*** specifying Metric P. Maybe it is something obviously silly, like the number of bug reports from each tester each week. Maybe it is a more subjective rating that considers the extent to which a tester achieves her scheduled objectives each week (adjusted by the ambitiousness and difficulty of the schedule and the achievement of other unscheduled but high priority tasks). Maybe it is a weekly "customer-satisfaction survey" completed by everyone who relies on the work product of the tester. Whatever it is, it is imperfect and potentially controversial.

As a starting analysis, we map the situation this way:

- ***Objective (Desired Outcome)***: Why are we computing Metric P (or anything like Metric P)? We'll start by asserting that the objective for computing Metric P is to improve the efficiency of the test group.

  The publicly stated objective might not be the underlying objective of the key decision makers. Different people involved in the decision to introduce this metric might have different objectives. For example, someone might say that their goal is to improve efficiency of existing staff but their actual goal might be to demonstrate that the workload is so heavy that no matter what efficiency improvements are made, more staff are needed. Someone else might intend to use Metric P to decide who to lay off and how many to lay off. A third person might want to identify tasks that should be reallocated into or out of the test group.

  - Reactions to the metric (and associated tensions) might depend heavily on *perceived* objective of different stakeholders.

- ***Subject.*** We'll start by considering the **tester** as the subject.

  It would be just as interesting to consider the test group (collectively), the test manager (you), the software development manager (or whoever is your group's primary customer), the executive most directly accountable for the performance of your group, and the manager of the corporate help desk (or whoever pays for support costs for the bugs that weren't found or weren't fixed). Each of these is a valid subject in this community, each has a stake in

the effectiveness of the metric used to assess tester productivity, each will suffer if the metric is unfair or unreasonable and thereby introduces chaos or rewards poor practices. Each of these viewpoints will lead us to different insights. But we have to start somewhere, so let's start with the individual tester.

- ***Object.*** We'll start by thinking about the program under test as the object that the test group is working on.

  The same thing can be sometimes an object and sometimes a tool. For example, when a tester is fully focused on writing a bug report, that report is the object that the tester is working on. The objective is to communicate effectively. Once the report is written, it becomes a tool. People use the information from the report to fix bugs or to prepare useful answers (if the bug is not to be fixed) for technical support (help desk) staff to provide to people who experience a failure caused by this bug.

  Similarly, the program under development is an object, but it becomes a tool when people start using it to help them do other things.

  Metric P might also have a dual role. To the extent that it is used to help the test group improve testing – this is the objective associated with using this metric – the metric is functioning as a tool. However, if the test group shifts attention – especially too *much* attention – to Metric P, it becomes an object. Metric P might also introduce dysfunction to the testing effort if the test group spends too much effort considering ways to move the metric in the right direction and not enough effort designing and running tests to improve the program.

    - We might also think of the test group as the object. The executive (the subject) wants to influence the test group (the object) using the new metric (the tool).

- ***Community***. If we start with the tester as the subject, the test group is the tester's most immediate community.

  People belong to several relevant communities. The tester associates with people outside of the test group, including other employees of the company (we could consider the entire company the community for the tester) and with groups (such as professional societies or trade associations) outside of the company.

  Note that we might think of different communities when we think about the subject (the milieu in which the subject lives and works), the object (if the company's product is the object, then user groups and regulators are part of the object-associated community), the tool, etc.

- ***Tools.*** We'll start by considering the new metric as a new tool.

  Is this tool good for its purpose? Is this a valid metric? A credible one? Does it measure what (some) people think it measures?

  We will also want to think about the other tools and how they might be affected by reliance on this metric. For example, will it incentivize the tester to rely more or less heavily on our new software test automation tool and why?

  Thinking of Metric P as a tool keeps our focus on the job of software testing where we use Metric P to improve our testing efforts in some way.

- ***Rules & Norms***. Legal rules, standards, societal customs, professional customs, and customs local to your organization or community all come into play here.

    - For example, what privacy rules govern the collection of data for Metric P? There might be constraints imposed by law and other constraints imposed by custom. Imagine an ambitious measurement advocate who wants to include, as part of his evaluation of the productivity of each staff member, how much time they spend in the bathroom and what they are doing that takes so long. Every aspect of the bathroom-data collection might be considered outrageous by your staff. Rules and norms are context-dependent. Suppose the object was the bathroom itself, the objective was to redesign the corporate bathrooms to make the use of them more likely to be pleasant, sanitary and less wasteful of employee time. Some of the customs that would bar study of bathroom use as a measure of productivity might be less offended by study of bathroom use as a measure of improvability of bathroom design.

  Coming back to Metric P, here's another example. Consider the community norms associated with test documentation. There is a heavyweight standard (IEEE 829) and a great deal of teaching about 829 in preparation courses for certification. If Metric P is positively influenced by results that are not directly benefitted by creating test documentation and if good documentation (or work spent creating good documentation) do not positively influence

the value of Metric P, then Metric P disincentivizes testers from creating this type of documentation. If the testers feel as though they have a professional (or legal) obligation to create such documentation, then Metric P is in direct conflict with documentation-related rules and norms.

Rules and norms don't always create conflicts. For example, in some companies, the staff (including the testers) often compare each other's work in terms of numbers of bugs found that week. In those companies, computing bugs per week as a measure of productivity would be entirely consistent with local norms even though (in our view) the long-term effect of using such a metric is toxic (Kaner, 1999).

- ***Division of Labor***. Introducing Metric P to the test group raises a variety of questions. Who will compute Metric P, who will create the Metric P report and who will interpret it? What effect, if any, will this focus on Metric P have on the testing effort? If testers (subjects) optimized their measured productivity (the objective) using Metric P, what tasks would they stop doing (insisting that other groups do them instead) and what improvements would they demand from other groups (such as better specifications or more extensive unit testing)?

Now consider some tensions:

- ***Primary tensions***: these are tensions within the same node. Here are a few examples:
    - ***Subject to subject***: Tester 1 has a working style that shows up favorably under Metric P whereas Tester 2 has a style that shows up less favorably. If Tester 2 thinks he is doing good work, what pressure does this put on him? What pressure does it put on his relationship with Tester 1?
    - ***Object to object:*** In our starting analysis, the program under test is the object. In what ways does Metric P influence the testers to actually improve the testing of the program? If the testers find more bugs this month but create tests that are less maintainable, is that a good tradeoff for this program or not? (For some programs, this would be an excellent tradeoff because there will be little or no future maintenance. For other programs, support for ports to new platforms, localizations, and for new versions might be critical.)
    - ***Community to community:*** Imagine a practice that is considered "best" by the American Society for Quality and "worst" by the Agile Alliance. If Metric P incentives or disincentives this practice, some members of the testing group will feel peer pressure (positive or negative) associated with Metric P.
    - ***Tool to tool:*** Suppose that Metric P counts bugs found per week, but with an expectation that many more bugs will show up early in testing than later. Early in testing, the emphasis on bug hunting might distract testers from developing regression test automation scripts or test documentation. To the extent that those are seen as valuable tools later in testing, they might not be available when they are needed.
- ***Secondary tensions:*** Between two nodes. Here are a few examples:
    - ***Subject to tool.*** Metric P devalues the work products that can be created with Tool (for example, a tool that models a software system in UML diagrams). However, Tester sees long-term career benefits in gaining expertise with Tool. Should Tester continue to work with the Tool or shift to something else?
    - ***Subject to tool.*** Consider Metric P as the tool. Whenever P incentives Tester to change behavior in ways that Tester does not want to change, there is a tension between the subject and the tool.
    - ***Object to community:*** Metric P incentives testers to focus on some tasks and not others. Suppose the expected result is a better quality product (the object is the program under test) but less documentation. Suppose the external user groups expect more documentation. By the way, if regulators expect more documentation, this would be a quaternary tension between the nearby activities of the company's software development and their regulatory obligations.
- ***Tertiary tensions:*** Between the old way and the new way. Here are a few examples:
    - ***"It is irresponsible to stop doing this task."*** Metric P incentivizes testers to refocus on some tasks at the expense of others. Sometimes, a test group has a long tradition of doing certain tasks and great pride in having done them at a certain level of quality.
    - ***"Our customers expect this."*** In the early 1980's, many software publishers stopped publishing lists of their error messages or known bugs because their competitors were using these lists against them. In turn, this reduced interest in making error messages unambiguous, tightly tied to specifically identifiable problems, and in testing the program to see whether the messages were correct and exhaustive. Testers still

did this type of work, but to the extent that the new Metric P incentivized them to do less of this work, testers might have complained that they were serving the product's customers less well.

- ***"This adds new work."*** The argument that it will take so long to collect Metric P data and write Metric P reports that any productivity benefits will be wiped out by the increased bureaucracy cost is an old-way-versus-new-way type of argument.

- ***Quaternary tensions:*** Between this activity system and its neighbors. Here are a few examples:
  - ***A litigation interface.*** The company produces products and wants to make better products that satisfy more customers at a lower development cost. It finds the perfect Metric P to incentivize testers to work toward this result. However a neighboring activity system involves litigation. When dissatisfied customers sue companies for bad products, it is useful to have certain types of records of the development process (and it may be useful to NOT have certain other types of records). Does Metric P optimize the output of this activity system (development of great products) at the expense of the important neighboring system (our lawyers defend us effectively in a litigation system)?
  - ***Standards interface.*** The company develops its products in the ways it sees as best, but some of their staff belong to professional societies who write industry standards or certification standards. The standards effort influences the company, and the company's internal standards influence the standards effort, via this interface (the staff).
  - ***Educational interface.*** Universities often teach a common view of software development: the "textbook way" of doing things. To the extent that Metric P incentivizes staff to do things differently, staff must be retrained. To the extent that a given university teaches a different approach that is more compatible with Metric P, companies that use Metric P will be more interested in recruiting graduates from that university. The university curriculum can influence the cost of recruiting and training and companies can influence the university curriculum.

Note that this analysis doesn't tell us whether Metric P is good or bad, or whether to adopt or oppose Metric P. It helps us see the implications of adopting Metric P. It helps us reduce the number of unexpected consequences of adopting Metric P.

Shifting the point of view (for example, looking through the eyes of different subjects) exposes new tensions and new benefits.

This is an open-ended analysis. The method does not provide a clean stopping rule. In practice, as you gain experience, you will probably stop when:

- You have gained enough insight to have an opinion about whether to adopt Metric P or not (if you are the decision-maker for this decision).
- You have gained enough insight to be able to present strong arguments and strong examples to the decision makers, about whether to adopt Metric P or how to modify Metric P to improve its benefits or reduce the toxicity of some of the tensions.
- You come up with the same issues over and over and not much new. This is a good time to bring some other people into your analysis to find out if they see new things (e.g. new nodes or new tensions), but at some point, there is just not much more that you can learn (even if, in theory, someone else somewhere else, could learn other things).
- You have run out of time. If the major issue is time boundary, we strongly suggest that you spread your available time over several days, so that you have a few opportunities to sleep on your thinking.

5.2 The New Tool

You are a senior member of the test group, focused on improving test group practices by improving the group's technology. You buy or create test tools, train staff in test techniques, and facilitate creation of new in-house testing standards (common practices within the company). Recently, you evaluated a new test tool and you believe that it can improve the work of your group. However, using it competently requires development of technical skills or subject-matter skills that some of the staff in your group do not have.

To make the example a little less theoretical, we will use a more specific example. Suppose that you are considering buying a tool that helps you describe the behavior of the program in terms of a state diagram and then designs an efficient series of state-model-based tests. For an extensive set of links to papers on model-based tests and tools, see Robinson (no date).

Here is some background. Imagine opening a word processing program with a blank document. The program has just created the document and is waiting for input. When the program is in this idle state, you might type some text into the document or you might bring up a preferences dialog and start configuring the program. When that dialog is open, you can't type new text into the document, but you can change the document's page margins. A state map would show, for each state that the program can enter, what other states the program can reach next (for example, from an idle state to pulling down a menu to bringing up a dialog). We would achieve 100% state-to-state coverage if we had a set of tests that put the program into each of its possible states and, from each state, tested all of the possible transitions from that state to another one.

- ***Objective (Desired Outcome)***: There are several *possible* objectives. Some people believe that the process of creating the model is likely to expose bugs. Others expect to find bugs with this style of testing that they would miss in traditional functional testing. Others want to maximize the efficiency of their state-based tests and expect the tool to choose the smallest set of tests that will achieve complete state-to-state coverage. Others still seek a rationale for selecting tests, i.e. a basis that seems less arbitrary or subjective than what they perceive as the norm among testers.

  Different people might adopt the same tool with different objectives. The members of the committee who agreed to buy the tool might each have their own objective and not much care for the objectives of the other committee members.

  For now, let's consider efficiency of achieving a high level of state-to-state coverage.

- ***Subject.*** The usual characters include the test, the test manager, and the project manager. In this case though, the technical writer who is charged with writing a user manual for the program might or might not find this tool useful. Similarly, the person or group who are writing the detailed functional specification might check the state-related logic of their specification (or the match of the program to their specification) using this tool. We might treat as a subject (analyze the project through the eyes of) anyone who stands to benefit from or be harmed by the new tool.

- ***Object.*** The software under test is the object. We might also think of the state model that the tool creates as an object. This is an output of the tool, but it is also something the tool uses to go a step further, creating a large suite of tests based on the model.

- ***Community***. One community includes the people developing the software under test. Another community, reflected at Robinson's website (Robinson, no date), are model-based testing enthusiasts. Another community might be a user group for the specific tool that generates a model-based testing set.

- ***Tools.*** The tool is the state-model-based testing tool that we are evaluating. Do we need other tools that interoperate with this one, to make this one work?

- ***Rules & Norms***. We have seen very few specifications that attempt a moderately thorough state map (or descriptions of control flow that could lead to a moderately thorough map). How compatible is this tool, and the state-transition research that this tool will demand, with the engineering culture of the group developing and testing this software? Alternatively, if this is a group that already likes state-transition models, how will the approach offered by the tool fit the norms of the company?

- ***Division of Labor***. Who will research the software under test to the extent needed to fill out the state map? Is this the task of the specification writer, the programmers, or the testers? Does Management imagine that the tool will generate a meaningful and correct map automatically?

Now consider some tensions. To save space, we will list a small number of examples per type of tension. However, in practice, analysis might surface dozens of tensions:

- ***Primary tensions***: these are tensions within the same node.
  - ***Subject to subject***: Consider the debates between testers who value the type of coverage that can be achieved using this tool and the testers who believe that other test techniques are more likely to expose a more diverse collection of bugs.
  - ***Object to object:*** The effort required to produce a high-quality state map (especially one that can be included in other documents) might conflict with the effort required to produce a high-quality program.
  - ***Community to community:*** The community of enthusiasts for this tool (or this style of testing) might offer testing priorities driven more by the style of testing than by the ultimate needs of the (community of) people who will use the software under test.

- ***Secondary tensions:*** Between two nodes. Here are a few examples:
    - ***Subject to tool:*** Suppose that a test tool is hard to use, hard to learn to use, and easy to forget how to use. The subject is the tester, working with the tool. This tool is unsuitable for someone who doesn't already have experience with it. A tester who is inexperienced is unsuitable for the tool. A tester who is experienced and skilled with the tool risks being stuck with it, ineligible for promotion or transfer until someone else has been painstakingly trained to work competently with the tool.
    - ***Tool to community:*** If the program fails in the middle of a long sequence of state transitions, what should the bug report look like? The report most readily available from the tool (run this thousand-step sequence and see this failure) might be almost useless to the community of maintenance programmers.
- ***Tertiary tensions:*** Between the old way and the new way. Here are a few examples:
    - ***"We never used to get bug reports like this."*** (See the "tool to community" secondary tension.) A bug report that gives an enormously long sequence of steps that eventually results in a failure will exasperate many debugging programmers because it might take them a long time to isolate the failure well enough to search effectively for the cause.
    - ***"Our old testing style focused on scenarios, on things that normal people would actually do. These tests are just unrealistic."***
    - ***"What do we put into the test case management system? How do we define a specific test case? Should we count a 500-transition test sequence as 1 test case? 500? "***
- ***Quaternary tensions:*** Between this activity system and its neighbors. For example:
    - ***Hiring:*** This approach is well respected but not yet widely used. It is seen as quite technical. Thus, it is difficult (and expensive) to find people already skilled in this approach and people on your staff who are now trained in the approach can probably find better-paying jobs somewhere else. (The neighboring systems are test groups in other companies.)

5.3 Describing failures persuasively

Testers report bad news.

Most testing reports are narrowly scoped: an individual bug report describes one specific problem.

Sometimes, however, testers pull together a collection of quality-related information, to write a broader assessment. For example, a tester might write a report that explains the implications (or costs) of a design decision, that highlights a pattern of design inconsistencies, that shows that the product's performance might look adequate on a few selected benchmarks but in normal use it responds much more slowly than those benchmark results suggest, or that an important group of intended users are likely to hate the product.

In many cases, there are political risks in writing such reports, and credibility risks.

- An example of political risk is that powerful people associated with the project might be angered by the report (and thus with the tester for reporting it).
- An example of credibility risk is that influential people associated with the project might dismiss the report because it expresses the opinion of a tester who is not qualified to pass judgment that a product is unusable, incompatible with market expectations, less responsive than the official performance studies suggest, etc.

It should be clear by now how CHAT might help you pull together materials (where to look for more materials and what additional questions / materials might be interesting) for a report like this. We want to add one further observation.

In writing a report, many testers emphasize their own assessment ("I think X is badly designed" or "I think users will hate X"). This maximizes their political and credibility risks. Imagine instead a report that features quotes from several stakeholders, in which they explain in their own words what they see as a problem or a weakness or a consequence of something done or not done or done too slowly or awkwardly by the software. The tester should provide a summary of the quotes, but even in that summary, the tone is that this is what these people are saying, not this is what this tester is thinking. This is often much more persuasive, much harder to dismiss, and to many readers it looks less like a personal attack than the shorter assessment written from the tester's point of view.

A CHAT-style analysis does not necessarily lead to detailed discussions with stakeholders, but if you have that type of data (such as technical support call records, magazine reviews, bug reports from users, and letters), CHAT will help you organize

it and your work with that data might suggest a few additional interviews to broaden your information base. It can also provide the structure for your report.

If you do include a lot of quotes in a report, it is important to quote representatively and accurately.

- You have a representative quote if it is one of many statements from many people that make the same point. You can accurately and fairly set the reader's expectations by pointing out that some statements are from just one person or one identifiable group of stakeholders. It will kill the credibility of the report (and of the reporter) if a quote is passed off as broadly representative when it is not.
- You have an accurate quote if it uses the actual words of the person quoted and presents the ideas of the person quoted fairly. It is often necessary to cut some words from a quote (repetitions or digressions), but show the cut in the quote and cut only in ways that preserve the meaning intended by the person. Similarly be careful to present the quote with enough context that it conveys the speakers message, not something that you are trying to force into a statement that was never intended your way.

In the *CommercialFolio* case study that we described above, we quoted extensively from real users. However, we synthesized those quotes (merged quotes from several people), in order to fit them into a relatively short conference paper. Our purpose in this paper is to give you a sense of the kinds of data that this style of work will bring to light. If we were writing a report on the software itself, or on the stakeholders at VendorBuilt College, we would have used the exact original quotes, without synthesis (as Fiedler, 2006, did). Misquotation is another way to kill your credibility.

5.4 Requirements analysis

The *CommercialFolio* case study illustrates the weaknesses of a product that was apparently designed without enough attention to the needs of one of the key users. This is hardly an isolated problem. Missing key stakeholders is one of the most widely acknowledged ways to fail in a requirements analysis (Lawrence, Wiegers, & Ebert, 2001). The diversity of stakeholders for typical software projects is enormous. Gause and Lawrence (1999) outline a process they use, that surfaces 200 or more stakeholder groups. This diversity is consistent with our software-related design, support and consulting experience.

A requirements documents is a list of things that some people think they want (Lawrence, 2000, a requirement is "any reason that drives design choices"). We might derive this list from other knowledge (such as our imperfect and incomplete perception of how people are currently doing a group of tasks), we might convince a roomful of people to sign a piece of paper that says they think they like the list too, and we might make it enforceable by writing a contract around it ("if you give me the things on this list, I'll pay you this amount of money"), but those things don't make the list correct or complete. Nor does a formalized representation of the list (Gause & Weinberg, 1989). We can work with those people to evaluate the list more carefully (Gause & Weinberg, 1989) but fundamentally conflicting interests among stakeholder groups are commonplace (Kyhlbäck & Sutter, 2007). Even in thoroughly reviewed projects, we often leave some requirements ambiguous because the key stakeholders are unwilling to reach agreement on these issues at this time and the project has to move forward (DeMarco, 1997).

Requirements analysis is one of the core activities of software testers. Think back to Jerry Weinberg's oft-quoted definition of software quality, "Quality is value to some person" (Weinberg, 1991, 1993). This carries a few important implications:

- Quality is subjective. What adds value for one person might reduce value for another. (This is just another way of describing the conflicts of interests and needs among stakeholders.)
- Anything that reduces the value of a product or service, for some person, reduces the quality of that product or service, to that person.
- A bug report, in its essence, is a report of a quality-related problem in the software. This might be a report of a coding error, but it might just as well be a report of an inadequacy of the product, a way in which the product is unnecessarily low in value to a user we care about. Some of *CommercialFolio*'s weaknesses in table-handling were there by design, not by coding error. But they were just as exasperating to the student users, and just as likely to reduce those users' willingness to pay for ongoing subscriptions to *CommercialFolio* after graduation.

From the quality-is-value perspective, we see a bug as very severe if it would cause a person to buy some other product instead, or to stop using this product after it has been acquired. It doesn't matter whether this is a coding error or a deviation from specification. What matters is that the software is of low value for an intended user (cf. Weinberg's definition of quality) and that the software is unfit for this person's use (cf. Juran & Gryna, 1988)

6. CONCLUSION

We are aware of the longstanding debate about the scope of testing. Some people see the role of the tester narrowly, as a person looking for coding errors or as a person looking for deviations from a thorough description of the intended product.

To them, it is out of the scope of the tester's work to determine who the important users (or other stakeholders) are for a product, what are their needs, interests, and potential irritants, what they will do with the product and what they expect from it (and on what basis), and then to assess the product against that analysis. We are on the other side of this debate. We call this the "not-my-job theory of testing" (Kaner, et al., 2001). We argue instead that the tester is one of the only people on the development project who has the opportunity to see and experiment with the entire product (Kaner, 1988) and that restriction of the testing role to the narrower view squanders a quality-critical opportunity. (For more discussion, see our course materials on bug advocacy: Kaner & Bach, 2005)

CHAT will not tell you how to identify all of the groups of people that a software product or service might impact. However, once you identify some, a CHAT analysis will help you discover other relevant aspects of their use to your product, discover their expectations, and imagine how the new software might affect them. CHAT has the potential to help you develop a rich understanding of the context in which your users live and work and in which your product will be used. Understanding the milieu surrounding your products users lets you explore how your product can delight or frustrate them as they engage in their activities.

7. ACKNOWLEDGMENTS

Many of the ideas in this paper were critiques at the 5th Workshop on Teaching Software Testing. We thank the participants at WTST 5 (James Bach, Jon Bach, Scott Barber, Michael Bolton, Steve Condly, Rebecca Fiedler, Morven Gentleman, Arthur Goldberg, Douglas Hoffman, Allen Johnson, Paul Jorgensen, Cem Kaner, David Klappholz, Panos Linos, Chang Liu, James Lyndsay, Darko Marinov, Pat McGee, Grigori Melnik, Robert Sabourin, and Andy Tinkham) for their ideas at WTST 5 and on many other occasions.

These notes are partially based on research that was supported by NSF Grant CCLI-0717613 “Adaptation & Implementation of an Activity-Based Online or Hybrid Course in Software Testing.” Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

4Teachers.org. (undated). RubiStar: Create Rubrics for your Project-Based Learning Activities. Retrieved July 1, 2009, from http://rubistar.4teachers.org

Appleton, B. (2000). Patterns and Software: Essential Concepts and Terminology. Retrieved July 1, 2009, from http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html

Batson, T. (2002). The electronic portfolio boom: What's it all about? *Syllabus Magazine* Retrieved October 9, 2003, from http://www.syllabus.com/article.asp?id=6984

Center for Activity Theory and Developmental Work Research. (n.d.). Cultural-Historical Activity Theory. Retrieved January 19, 2006, from http://www.edu.helsinki.fi/activity/pages/chatanddwr/chat/

Cohn, E. R., & Hibbitts, B. J. (2004). Beyond the electronic portfolio: A lifetime personal web space. *Educause Quarterly, 27*(4), 7-10.

DeMarco, T. (1997). *The Deadline*. New York: Dorset House.

Engeström, Y. (1987). Learning by Expanding: An Activity-Theoretical Approach to Developmental Research. Retrieved February 12, 2005, from http://lchc.ucsd.edu/MCA/Paper/Engestrom/expanding/toc.htm

Engeström, Y. (2000). Activity theory as a framework for analyzing and redesigning work. *Ergonomics, 43*(7), 960-974.

Engeström, Y., & Blackler, F. (2005). On the life of the object. *Organization:, 12*(3), 307-330.

Fiedler, R. L. (2006). *"In transition": An activity theoretical analysis examining electronic portfolio tools' mediation of the preservice teacher's authoring experience.* Unpublished Ph.D. dissertation, University of Central Florida (Publication No. AAT 3212505).

Fiedler, R. L. (2007). *Portfolio authorship as a networked activity.* Paper presented at the Society for Information Technology and Teacher education

Foot, K. (2002). Pursuing an evolving object: A case study in object formation and identification. *Mind, Culture and Activity, 9*(2), 132-149.

Gause, D. C., & Lawrence, B. (1999). User-driven design. *Software Testing & Quality Engineering, 1*, 23-27.

Gause, D. C., & Weinberg, G. M. (1989). *Exploring Requirements: Quality Before Design*. New York: Dorset House.

Gay, G., & Hembrooke, H. (2004). *Activity-Centered Design: An Ecological Approach to Designing Smart Tools and Usable Systems*. Cambridge, MA: The MIT Press.

Giest, H. (2008). The formation experiment in the age of hypermedia and distance learning. In B. Van Oers, W. Wardekker, E. Elbers & Van der Veer (Eds.), *The Transformation of Learning: Advances in Cultural-Historical Activity Theory* (pp. 100-126). Cambridge: Cambridge University Press.

Halverson, C. A. (2002). Activity theory and distributed cognition: Or what does CSCW need to do with theories? *Computer Supported Cooperative Work (CSCW), 11*(1), 243-267.

Hoffmann, M. H. G. (2005). The curse of the Hegelian heritage: "Dialectic", "contradiction", and "dialectical logic" in activity theory. *Georgia Tech: Ivan Allen College School of Public Policy Working Paper Series, Working Paper #9*. Retrieved from http://www.spp.gatech.edu/faculty/workingpapers/wp9.pdf

Holt, G. R., & Morris, A. W. (1993). Activity Theory and the analysis of organizations. *Human Organization, 52*(1), 97-109.

Holzman, L. (2006). What kind of theory is activity theory? *Theory and Psychology, 16*(1), 5-11.

Juran, J. M., & Gryna, F. M. (1988). *Juran's Quality Control Handbook* (4th ed.). New York: McGraw-Hill.

Kaner, C. (1988). *Testing Computer Software* (1st ed.). New York: McGraw Hill.

Kaner, C. (1999, May/June). Don't Use Bug Counts to Measure Testers. *Software Testing & Quality Engineering*, 79-80.

Kaner, C. (2009, June 10, 2009). What is context-driven testing? http://www.satisfice.com/kaner/?p=45

Kaner, C., & Bach, J. (2005, January 17, 2007). Black Box Software Testing: A Video Course. Retrieved July 1, 2009, from http://www.testingeducation.org/BBST

Kaner, C., Bach, J., & Pettichord, B. (2001). *Lessons Learned in Software Testing*: Wiley.

Kaptelinin, V., & Nardi, B. A. (2006). *Acting With Technology: Activity Theory and Interaction Design*. Cambridge, MA: The MIT Press.

Kuutti, K. (1995). Work processes: Scenarios as a preliminary vocabulary. In J. M. Carroll (Ed.), *Scenario-Based Design: Envisioning Work and Technology in System Development* (pp. 19-36). New York: John Wiley & Sons.

Kuutti, K. (1996). Activity theory as a potential framework for human-computer interaction research. In B. A. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge, MA: MIT Press.

Kuutti, K., & Arvonen, T. (1992, October 31-November 4). *Identifying potential CSCW applications by means of activity theory concepts: A case example.* Paper presented at the CSCW'92 Conference on Computer-Supported Cooperative Work, Toronto.

Kyhlbäck, H., & Sutter, B. (2007). What does it take to replace an old functioning information system with a new one?: A case study. *International Journal of Medical Informatics, 76*(Supplement 1), S149-S158.

Laszlo, E. (1996). *The Systems View of the World: A Holistic Vision for Our Time (Advances in Systems Theory, Complexity, and the Human Sciences)*. Cresskill, NJ: Hampton Press.

Lawrence, B. (2000). *Rethinking requirements*. Paper presented at the International Conference on Software Testing Analysis & Review (STAR West 2000). Retrieved from http://www.stickyminds.com/s.asp?F=S5470_CP_2

Lawrence, B., Wiegers, K., & Ebert, C. (2001). The top risks of requirements engineering. *IEEE Software, 18*(6), 62-63.

Nardi, B. A., & Engestrom, Y. (1998). A Web on the Wind: The Structure of Invisible Work: Special Issue of Computer Supported Cooperative Work. *Computer Supported Cooperative Work, 8*(1-2).

Nardi, B. A., & Redmiles, D. F. (2002). Activity Theory and the Practice of Design: Special Issue of Computer Supported Cooperative Work. *Computer-Supported Cooperative Work, 11*(1-2).

PortlandPatternRepository. (undated). CategoryPattern. Retrieved July 2, 2009, from http://c2.com/cgi/wiki?CategoryPattern

Robinson, H. (no date). Model-Based Testing Home Page. Retrieved June 10, 2009, from http://www.geocities.com/model_based_testing/

Russell, D. (1997). Rethinking genre and society: an activity theory analysis. *Written Communication, 14*(4), 501-551.

Seaman, J. (2008). Adopting a grounded theory approach to cultural-historical research: Conflicting methodologies or complementary methods? *International Journal of Qualitative Methods, 7*(1), 1-17.

Shulman, L. (1998). Teacher portfolios: A theoretical activity. In N. Lyons (Ed.), *With portfolio in hand: Validating the new teacher professionalism* (pp. 23-37). New York: Teachers College Press.

Von Bertalanffy, L. (1976). General System Theory: Foundations, Development, Applications. New York: George Braziller.

Weinberg, G. M. (1991). *Quality Software Management: Systems Thinking* (Vol. 1): Dorset House.

Weinberg, G. M. (1993). *Quality Software Management: First-Order Measurement* (Vol. 2): Dorset House.

Weinberg, G. M. (2001). *An Introduction to General Systems Thinking (Silver Anniversary Edition)*. New York: Dorset House.