Open Courseware & Open Certification in Software Testing

Cem Kaner, J.D., Ph.D.

IQAA October, 2006

Copyright (c) Cem Kaner 2006. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/2.0/ or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

These notes are partially based on research that was supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

I think our field has gotten stuck. Programmer productivity has escalated by orders of magnitude in the past two decades. Ours hasn't. Today's dominant test techniques were developed in the 1970's, back when big programs were 10,000 statements. Today, big programs are 10 million. The productivity differential means that we impact the program less every year. Unless this changes, we will become irrelevant. In the visions of some methodologies, we already are.

One important way that change propagates is through the educational system. For example, in my courses on introductory programming and programmer testing, I require my students to do test-first programming. They carry the ideas to co-workers or to their students. Employers who like the change press other staff (and other schools) for more of this type of work. Eventually, like object-oriented programming, it becomes the norm. We have no support structure like that for testing. Testing education at university is limited (few courses), generally shallow, and often theoretical in a way that reinforces learning of the core computer science concepts these students are learning in their other courses. If we need to foster a new generation of test architects--as I think we do--the traditional school system will help by fostering much knowledge of architecture, technology, and implementation planning, but it will probably not foster much wisdom about testing per se.

The commercial training system helps less. We have courses today -- and certifications that are sold with them -- that reinforce the style of the 1980's. Short courses can help introduce new ideas, but they are too time-constrained to support development of skill or critical reflection on the concepts under study.

As to current certification exams, where is the measurement of skill, application of ideas to complex situations, or evaluation of competing approaches? Knowledge of basic definitions and simple applications provides minimal (or no) indication of the level of knowledge needed to actually do testing work well. In my view, to the extent that employers believe that current certificates measure knowledge or skill in the field, they are fools.

For the past 6 years, I have been developing an approach that blends academic and commercial instruction. It still needs work. Course development is expensive and I have only one course developed this way. But I think it shows promise as an approach that can help us grow more sophisticated staff than the academic/commercial education systems have been helping us grow today. In coordination with that, some of us have been working on new models for certification. We have ideas toward version 1.0, work in progress, and a challenge in principle to some (not all) of the other certifiers.

Source Materials on the Disk

- Video lectures
- Activities
- Assignments
- Sample exam questions
- Some readings
- A little additional instructional support material
- These materials echo sites that James Bach, Scott Barber, Tim Coulter, Rebecca Fiedler and I have been creating at Florida Tech (www.testingeducation.org) and Satisfice (www.satisfice.com) that will give access to reusable content and host supervised courses. (Some of the Satisfice-site courses will cost money, others will be free.)
- All of my instructional materials are available, royalty-free, under the Creative Commons license.

- Most of today's software testing techniques were developed in the 1970's.
 - Back then, long programs were 10,000 statements
 - Code was often readable COBOL
 - An enterprising tester could read the entire program, identify all the variables and most of the relevant combinations.

- Over the past few decades, programmer productivity has surged, driven by revolutions in software engineering practice.
- Class libraries make it easy to snap together large applications
- Many consumer products include millions of lines of code.

- We have not experienced revolutions in testing practice and we are not much more productive today than we were three decades ago:
 - Regression test automation offers small, incremental improvements in productivity
 - High-volume test automation is still rarely done and is poorly understood by the general (e.g. academic) testing community
 - Test case documentation is as overblown as ever, with a new generation of semi-automated "test-case management" bureaucracy to slow us down further.

- We are much better at
 - testing
 - documenting the testing
 - reporting status of the testing of a 10,000 statement program.
- But as the size of programs grows geometrically
- and the efficiency of testers grows maybe linearly
- we impact less of the program every year.
- System-level testing will become irrelevant, because we will impact so little of the product.

Alternatives to Consider

- Maybe black-box, system-level testing is obsolete
- Maybe certifications will assure skill in our field
- Maybe university training in testing will foster skill and the evolution and spread of new paradigms

Maybe we have to revolutionize commercial training

Maybe Black-Box System-Level Testing is Obsolete

- It is wildly inefficient to expose unit-level bugs (like input-field filter bugs) with system-level tests.
- However, many aspects of software behavior emerge in the broader system, not at the unit level:
 - Race conditions
 - Stack corruption, memory leaks, odd error handling
 - Utility, security, performance, etc.

Maybe Black-Box System-Level Testing is Obsolete

• *If*

Quality is value to some person

--Weinberg

- Then we still need to investigate the extent to which the product under development provides or fails to provide appropriate value to the relevant stakeholders.
- Unit testing doesn't address this
- Simplistic functional testing, including "story" testing, barely begins to address this.

Maybe Black-Box System-Level Testing is Obsolete

The need for this type of investigation is still present.

Whether we can competently satisfy that need remains to be seen.

Maybe Certification Will Assure Skill

- Current certification exams focus on superficial knowledge (e.g. definitions, memorizable descriptions, etc.) rather than evaluation of skilled performance
- Review courses that teach you how to pass the certification exam are very commercially successful at the moment, but I question their educational value (more later in these slides).
- The "bodies of knowledge" that I've reviewed seem firmly grounded in 1970's/1980's material.
- I don't see how this moves us forward

Maybe university training in testing will foster skill and evolution / spread of new paradigms

- Universities have played a large role as change agents in other fields (including programming & design)
 - New development paradigms turn into new courses or rewrite old ones:
 - UML is mainstream
 - OO design is mainstream
 - Test-first programming is spreading in early courses
 - New graduates infiltrate new approaches into traditional workgroups

Maybe university training in testing will foster skill and evolution / spread of new paradigms

- We considered a testing degree at Florida Tech
 - Rich enough intellectual problem to deserve a degree.
 - But serious risk of career stereotyping and lock-in
 - Abandoned in favor of a software engineering degree that offers
 - Courses in black box software testing and programmer testing
 - additional test-relevant courses (e.g. human factors) and testing options.
 - This is better than most other places but still just scratches the surface.

Maybe we have to deal with it as industrial training

- Industrial training has its own challenges
- I left a very successful consulting practice
 - whose main income generator was commercial training
 - accepting a 2/3 reduction in income
 - because I concluded that most commercial training is good for introducing new concepts but not good for developing skills or critical insight.
- If we're going to foster deeper skill development and richer evaluation of practice if we want to kickstart the next productivity revolution in testing the short-course model won't do it.

Commercial vs. Academic

- Drive-by teaching
 - 2-5 days, rapid-fire ideas,
 visiting instructor
- Broad, shallow coverage
- Time constraints limit activities
- No time for homework
- No exams
- Coached, repeated practice seen as time-wasting
- Familiarity
- Work experience helps to bring home concepts
- Richer grounding in real practice
- Some (occasional) student groups share a genuine current need
- Objective: one applicable new idea per day

- Local teaching
 - Several months, a few hours per week, students get to know instructor
- Deeper coverage
- Activities expected to develop skills
- Extensive homework
- Assessment expected
- Coached, repeated practice is highly appreciated
- Capability
- Students have no work experience, need context
- Harder to connect to real practice
- Students don't naturally come to a course as a group with a shared problem
- Expect mastery of several concepts and skills

My idea has been to develop courses in an academic environment (where I can learn more about what works and why), with the goal of providing an alternative model for commercial (in-house) training and professional self-study

Today's workshop is a progress report against a broader curricular vision

Summary so far

- Testing has evolved slowly compared to related fields
- Educational opportunities for testing are limited
- University-level educational opportunities will continue to be limited
- Conclusion:
- You want a good tester?
 - Plan A: Hire experienced staff who can demonstrate relevant knowledge, skill and attitudes
 - Plan B: Train your own staff
- This course is about Plan B.

Overview

- Tour of the Moodle course management system
- Tour of the Black Box Software Testing Course on Moodle
- Overview of use of material like this in the workplace
- Dealing with the instructional challenges of a cognitively complex field of study

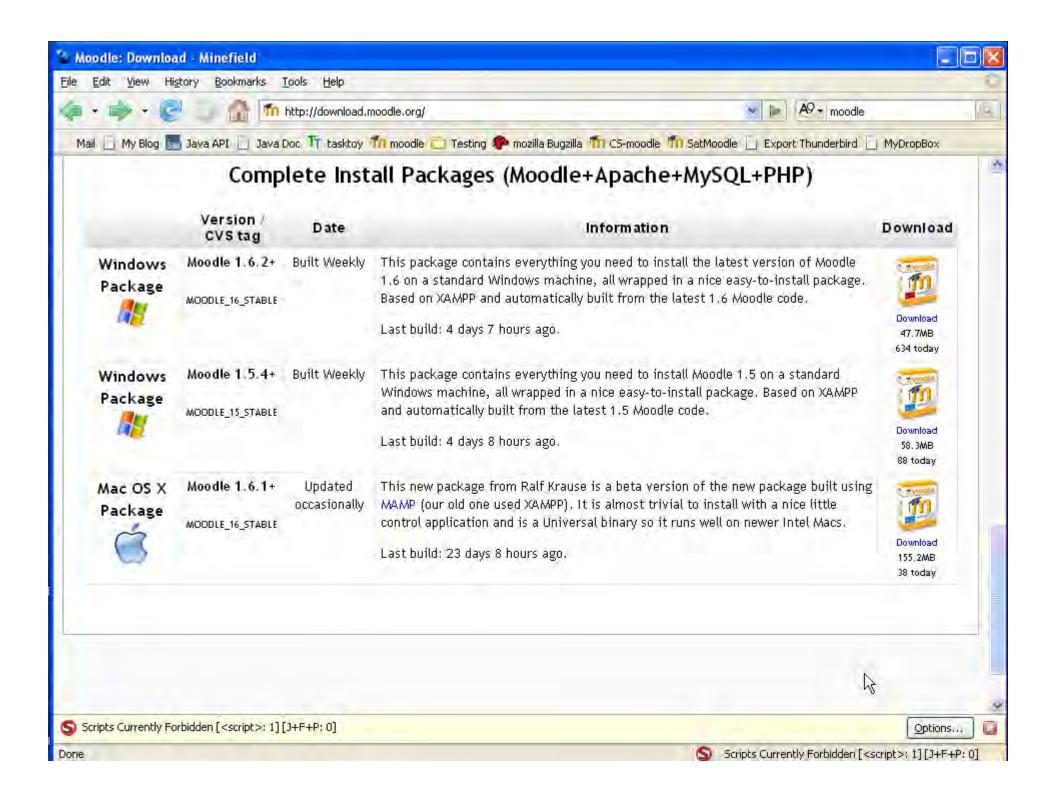
An Overview of Moodle

- www.moodle.org
- Free
- Course management system
- Useful for:
 - Live short courses (requires web access)
 - Live academic courses (long term, homework)
 - Hybrid of remote / live
 - Remote courses
 - Synchronous (live web conference tools are better)
 - Asynchronous



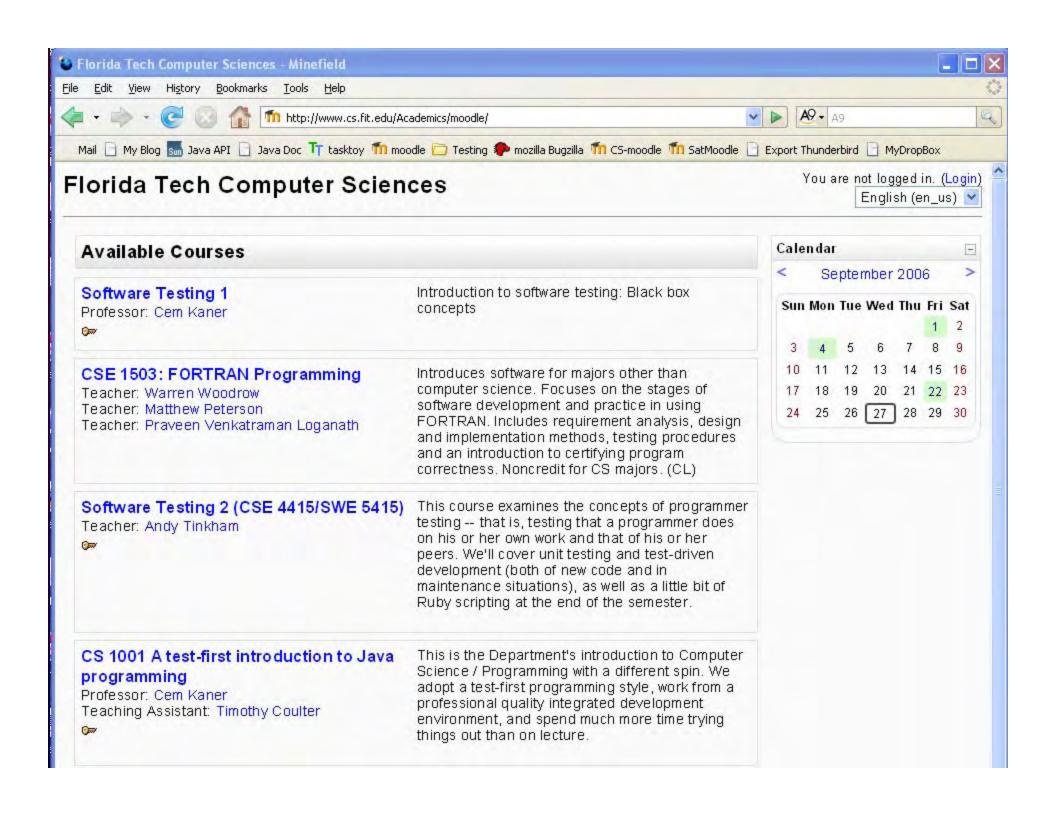
Moodle platforms

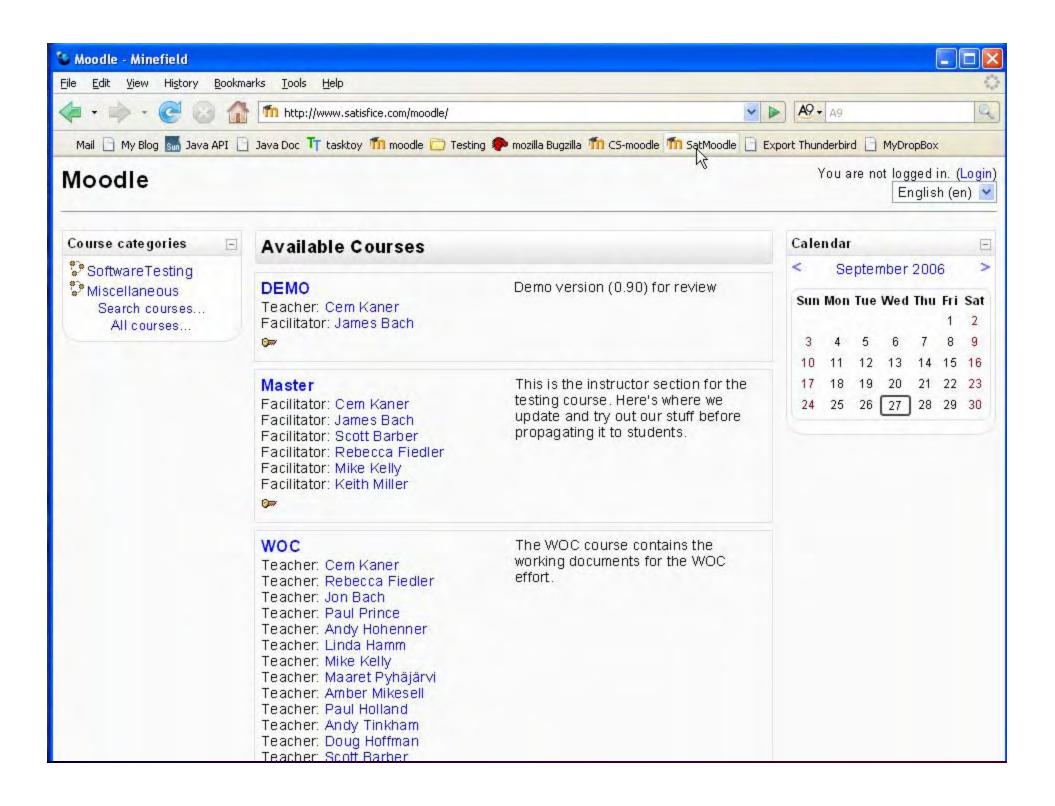
- Windows
- Mac
- Linux

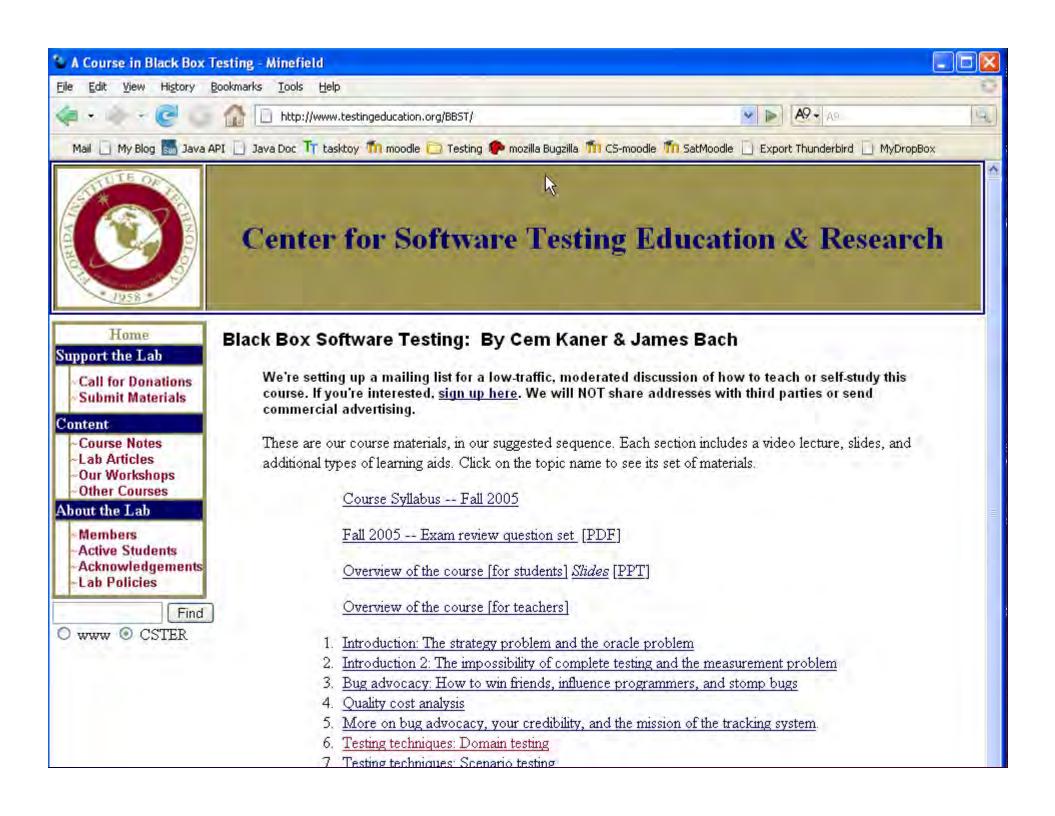


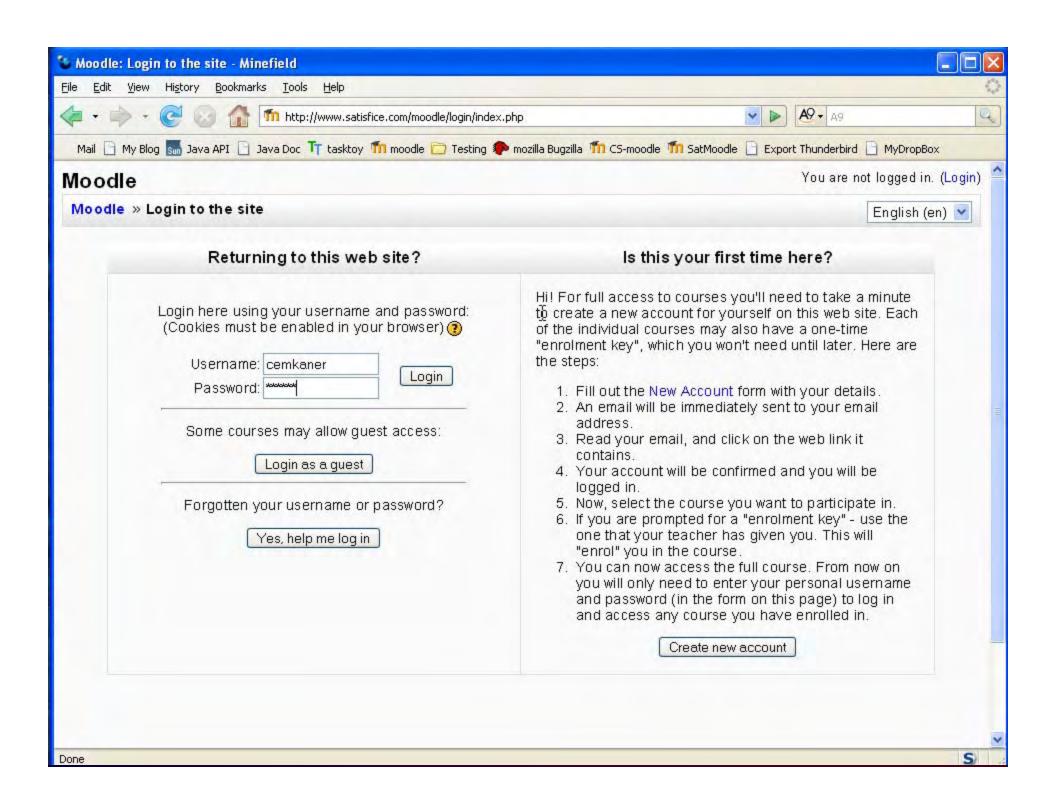
Overview of Moodle

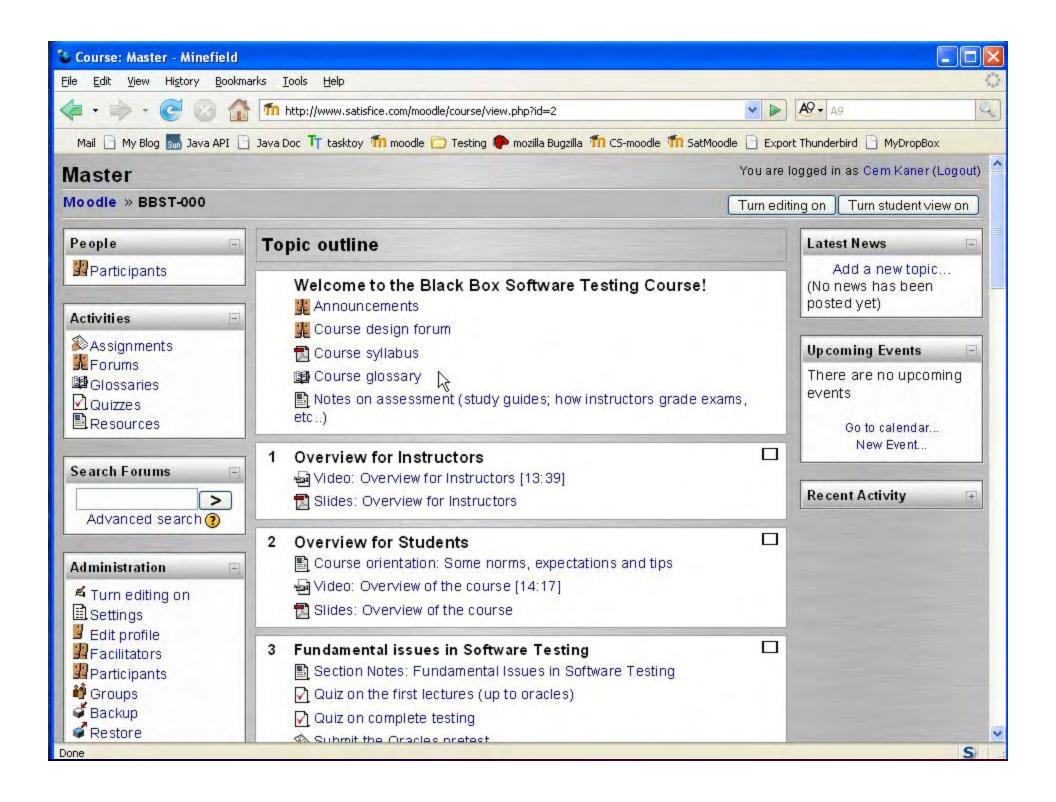
- The following are samples from some courses / activities that I host on moodle
- Some data / demonstrations are unavailable (e.g. layout of quiz results) because of student confidentiality rules

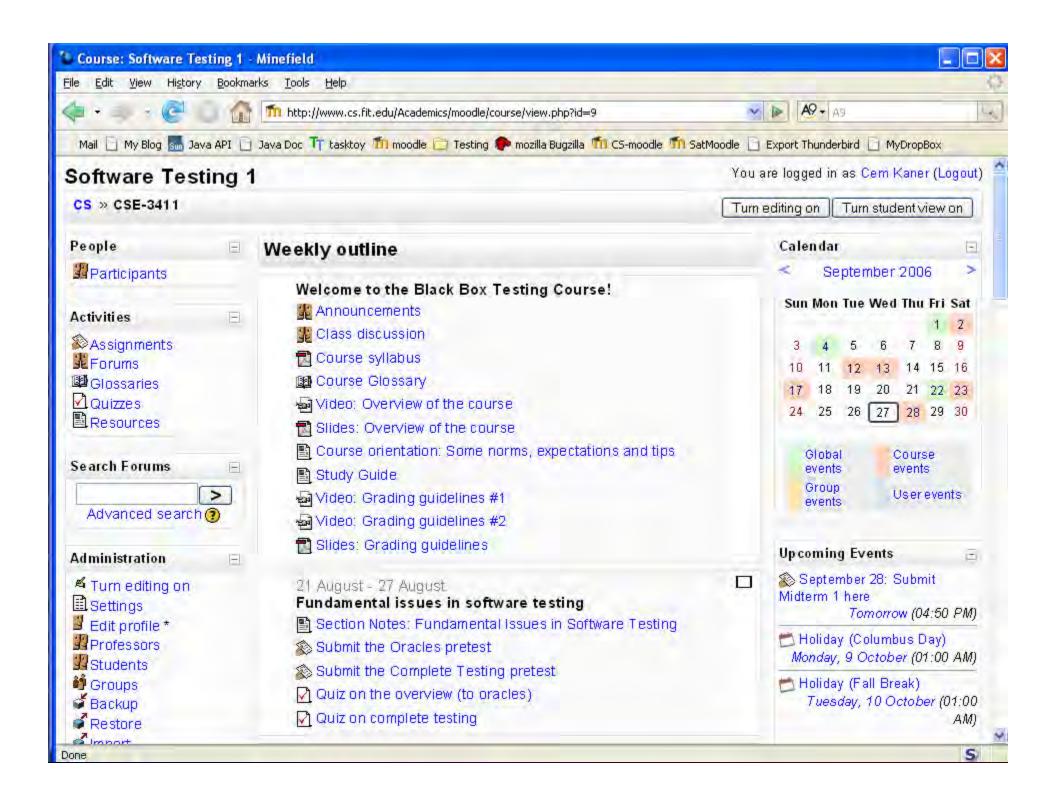


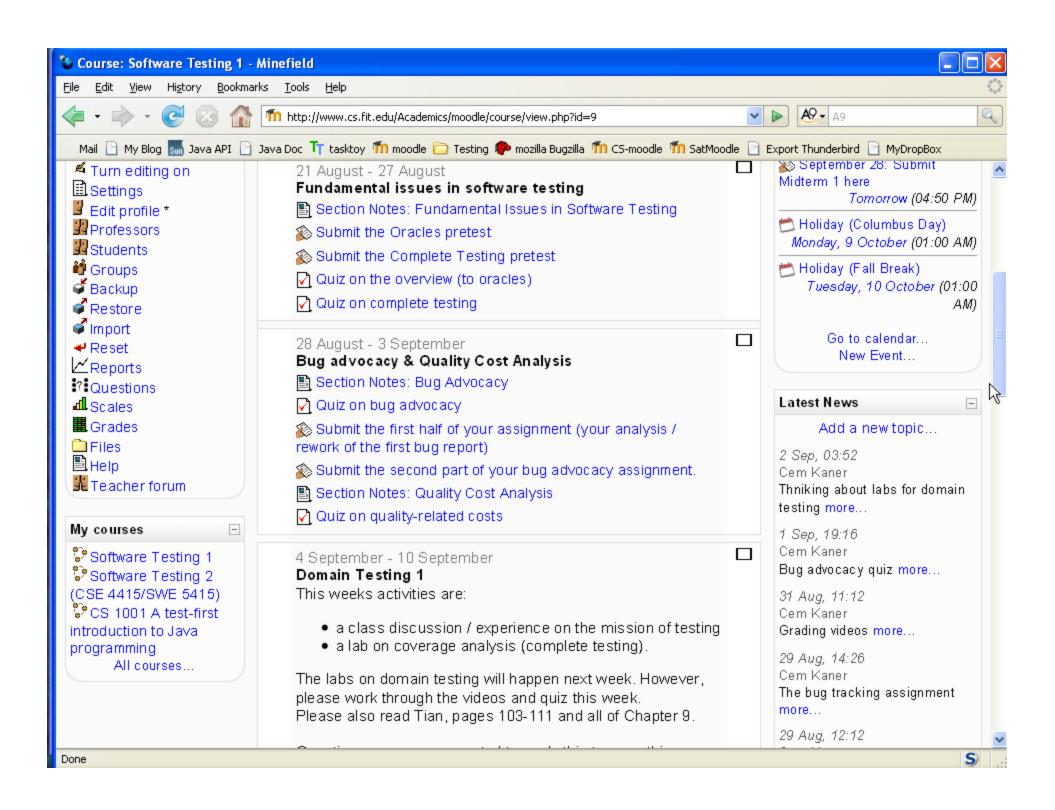


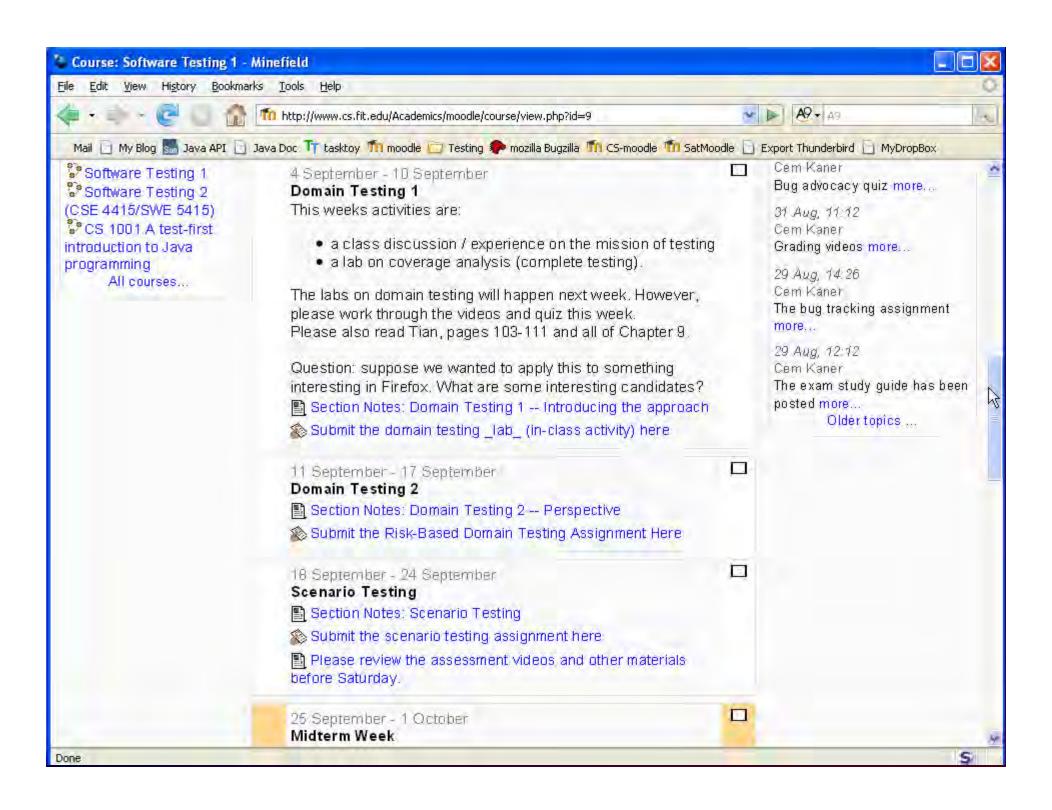


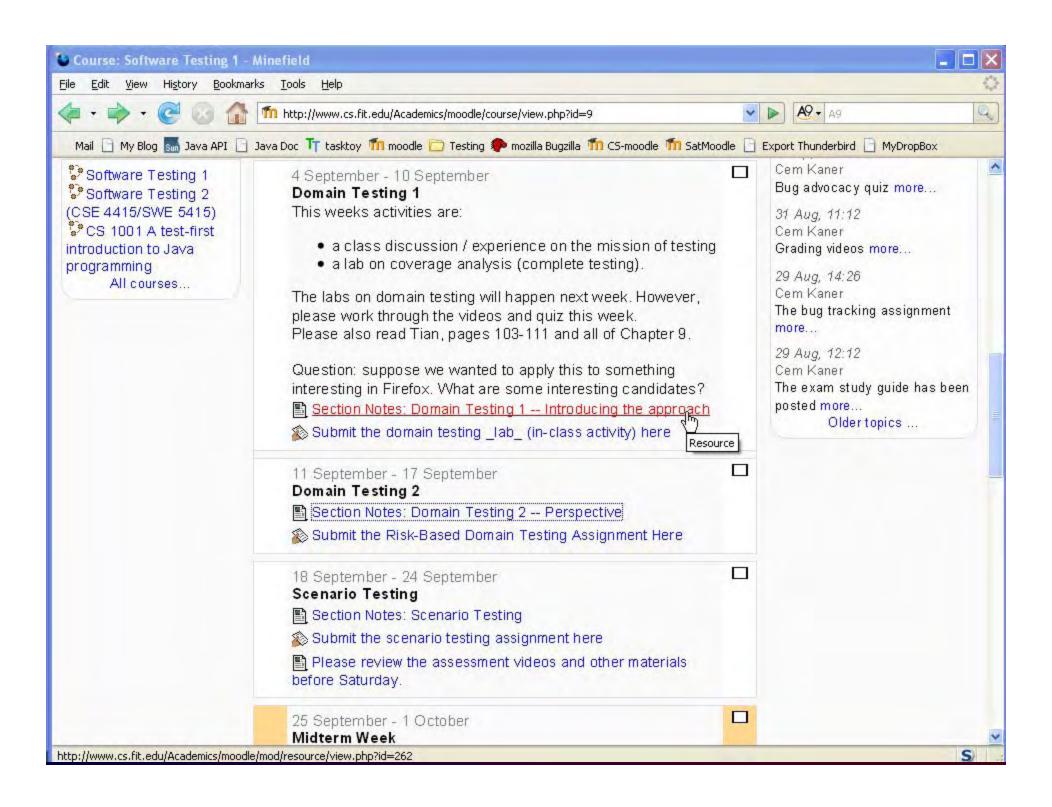


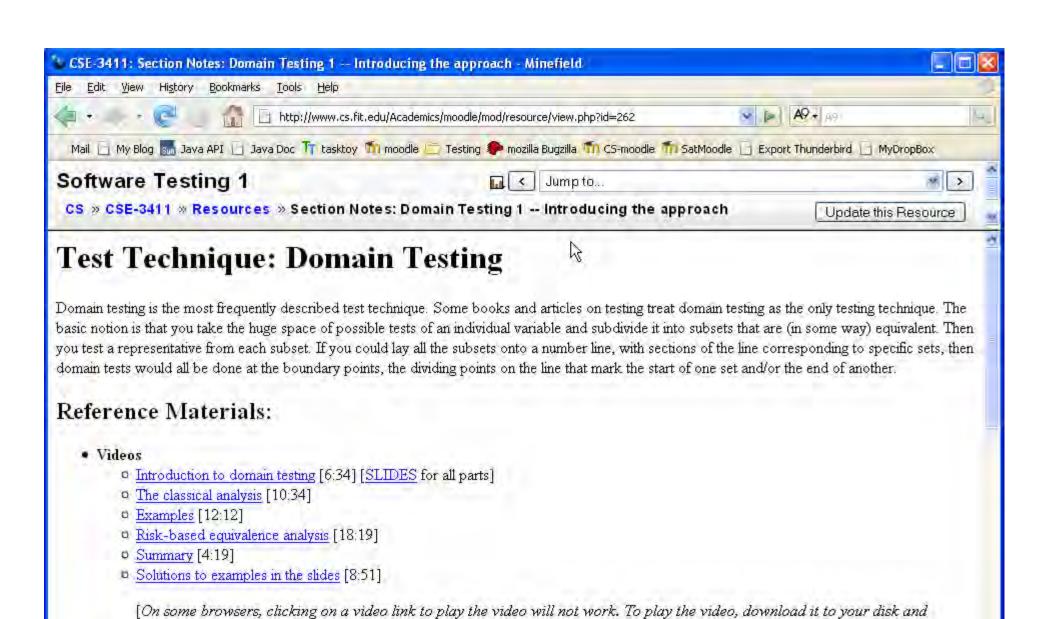








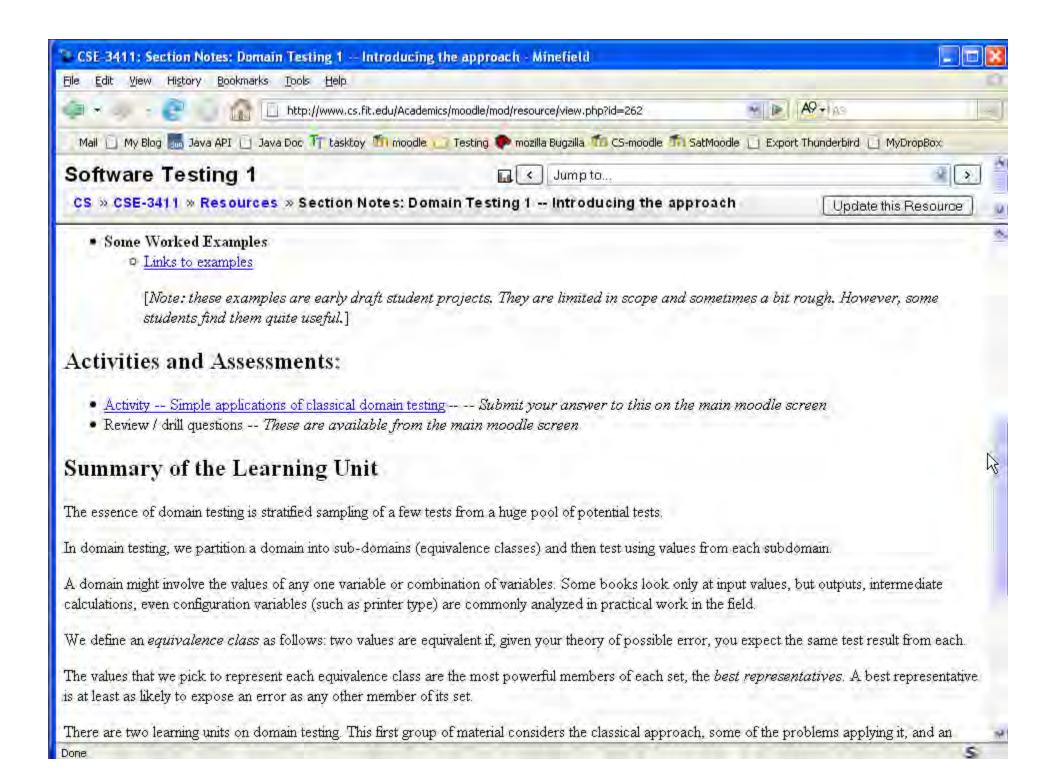


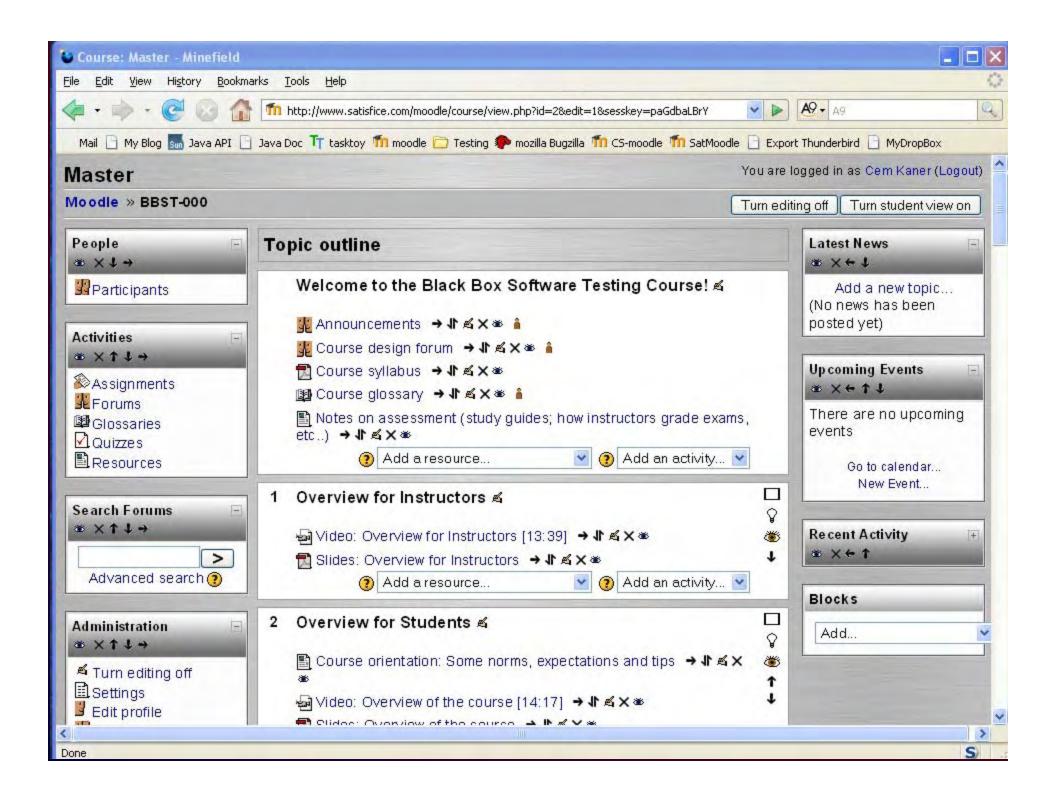


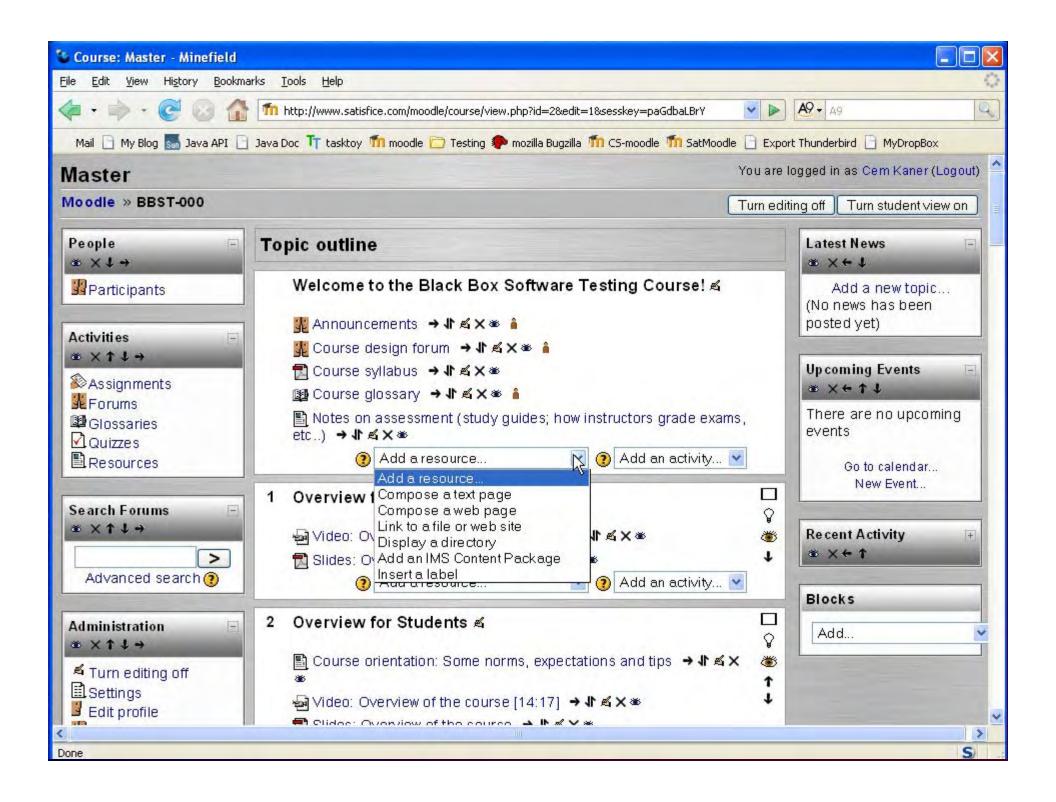
Ostrand and Balcer, The Category-Partition Method for Specifying and Generating Functional Tests. CACM June 1988 (31:6)

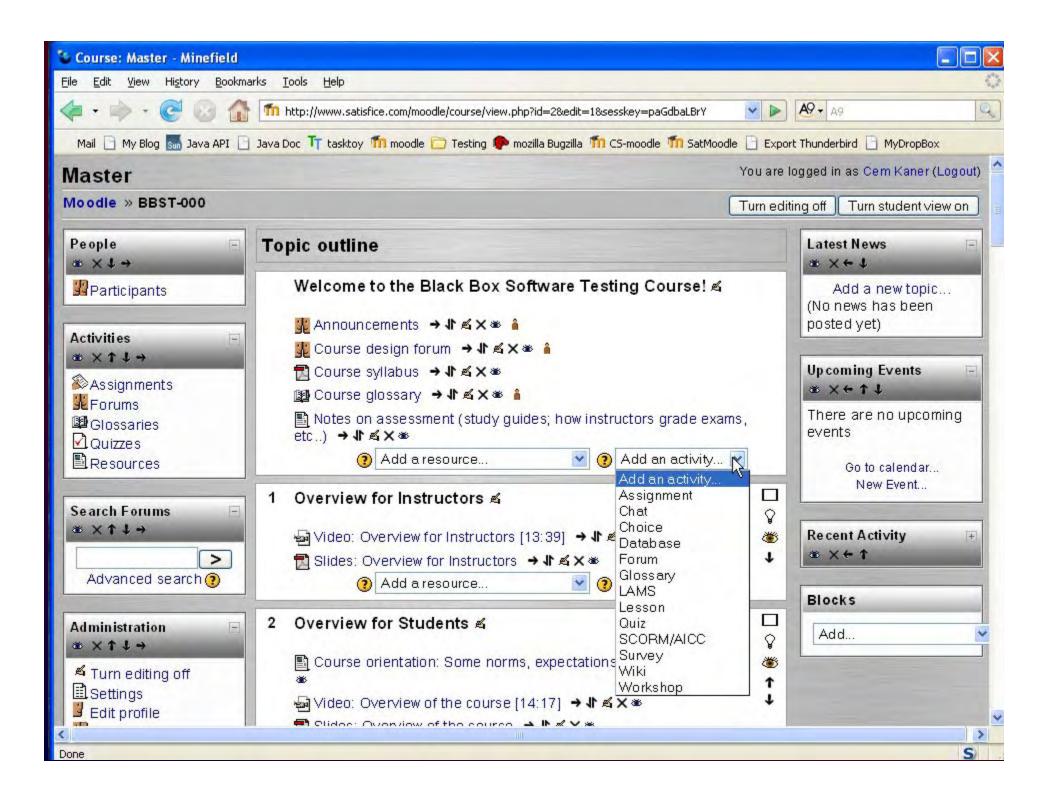
play the downloaded copy with Windows Media Player 9 or later.]

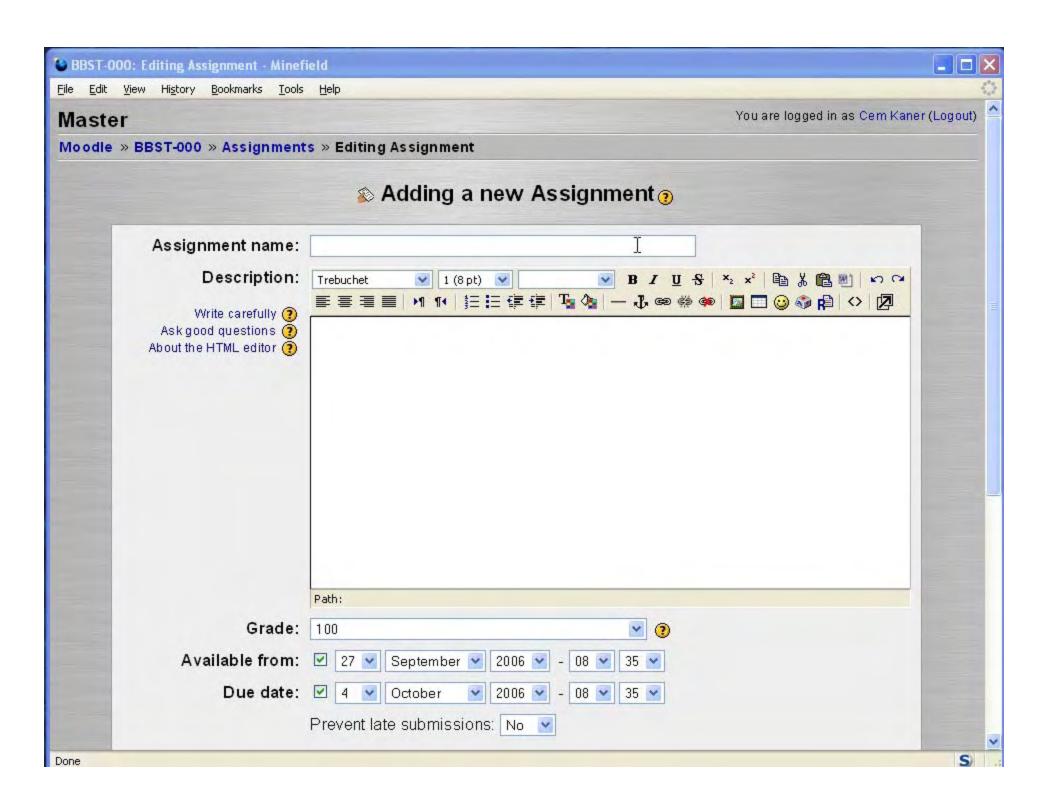
Some Worked Examples

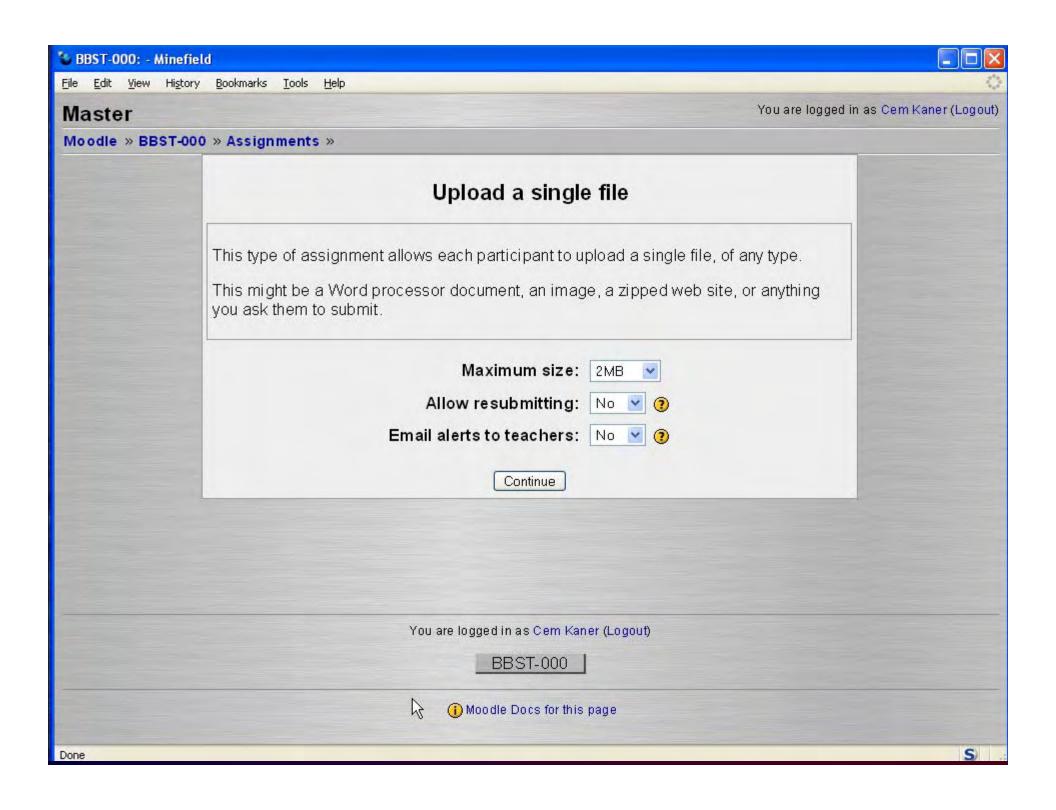


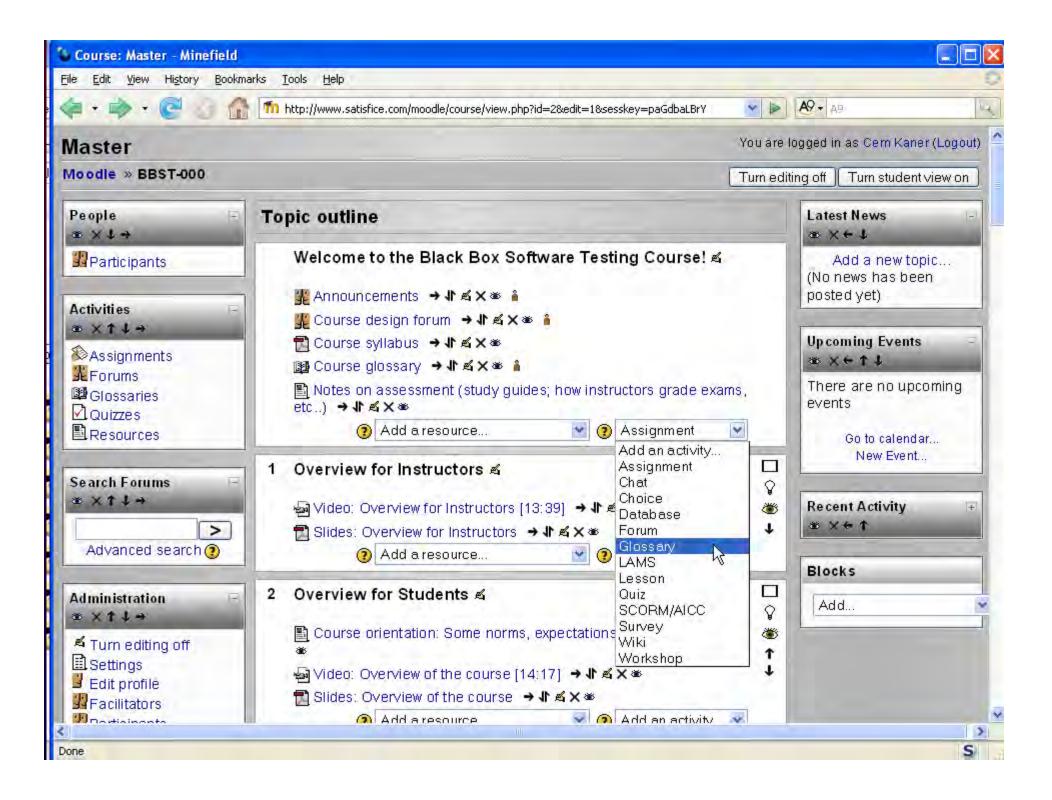




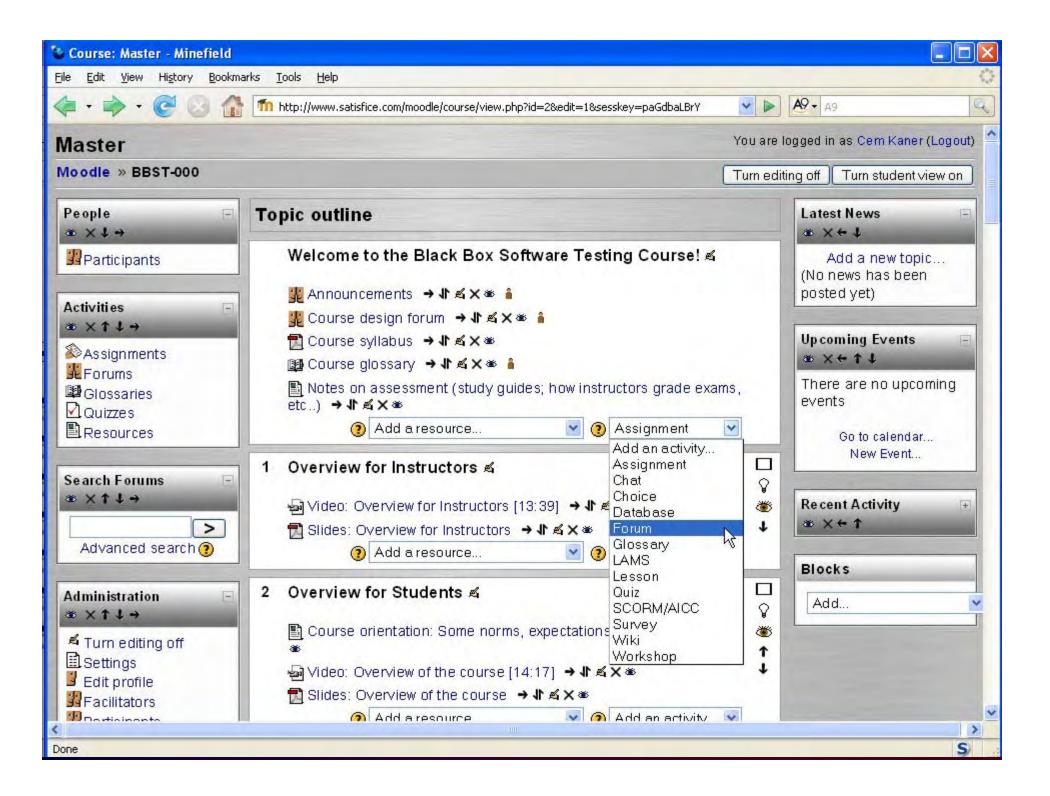


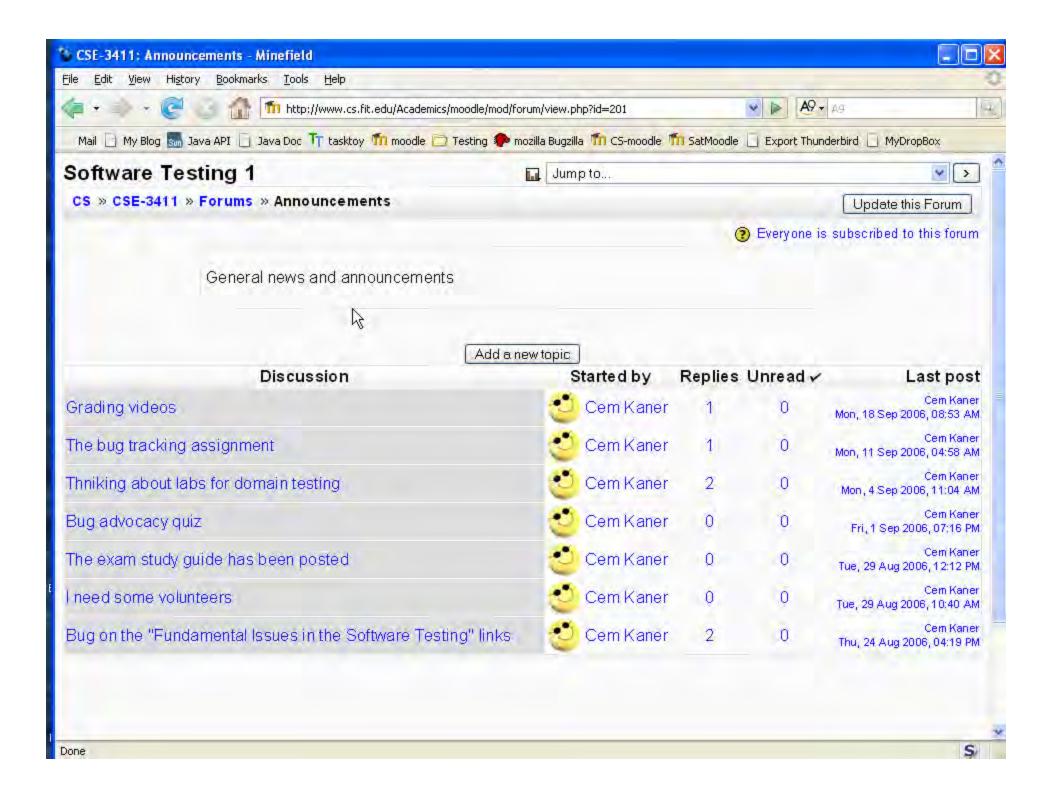


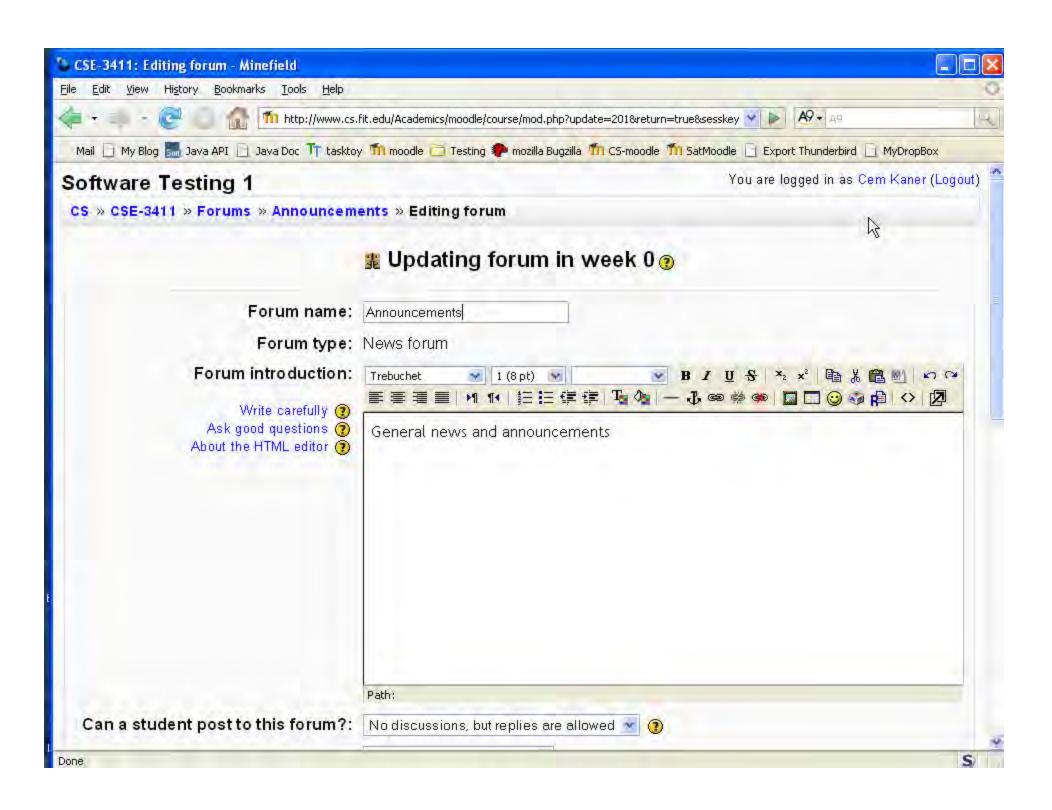


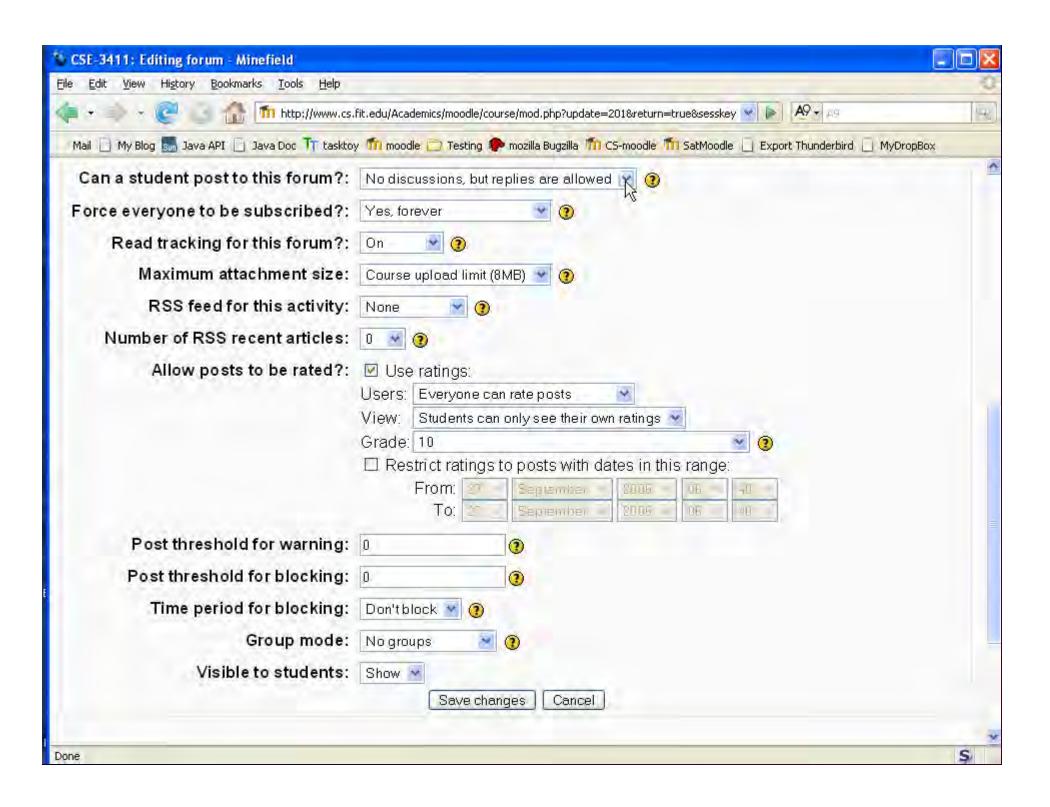


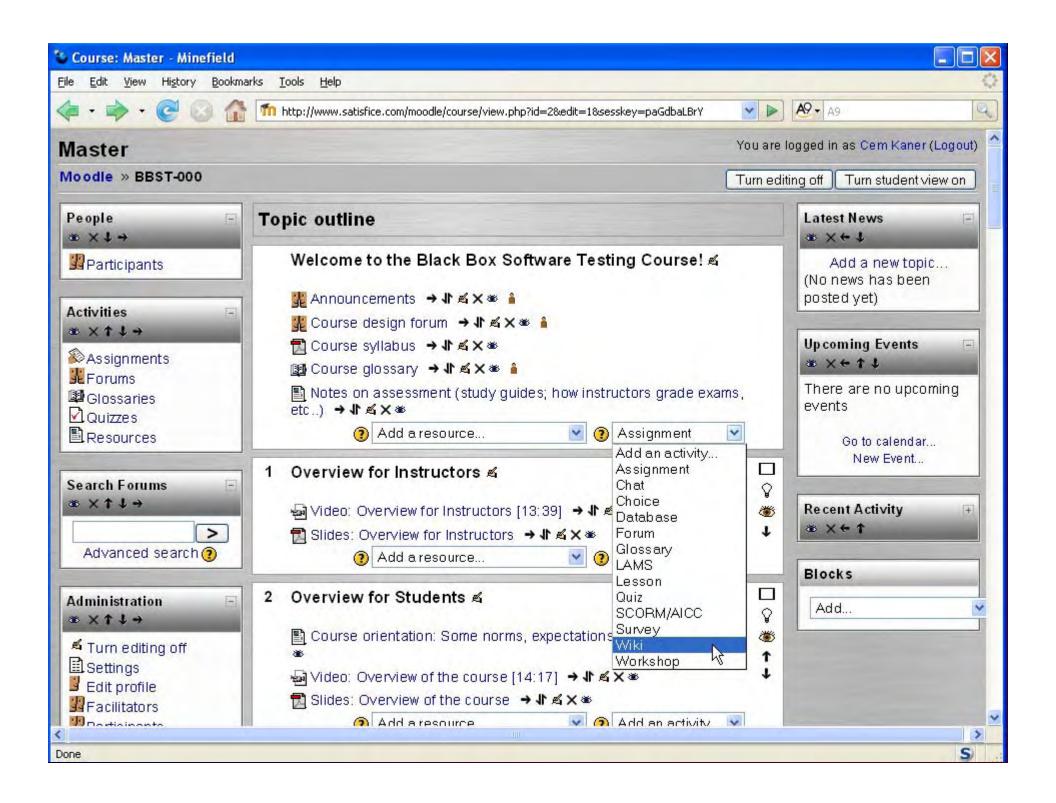


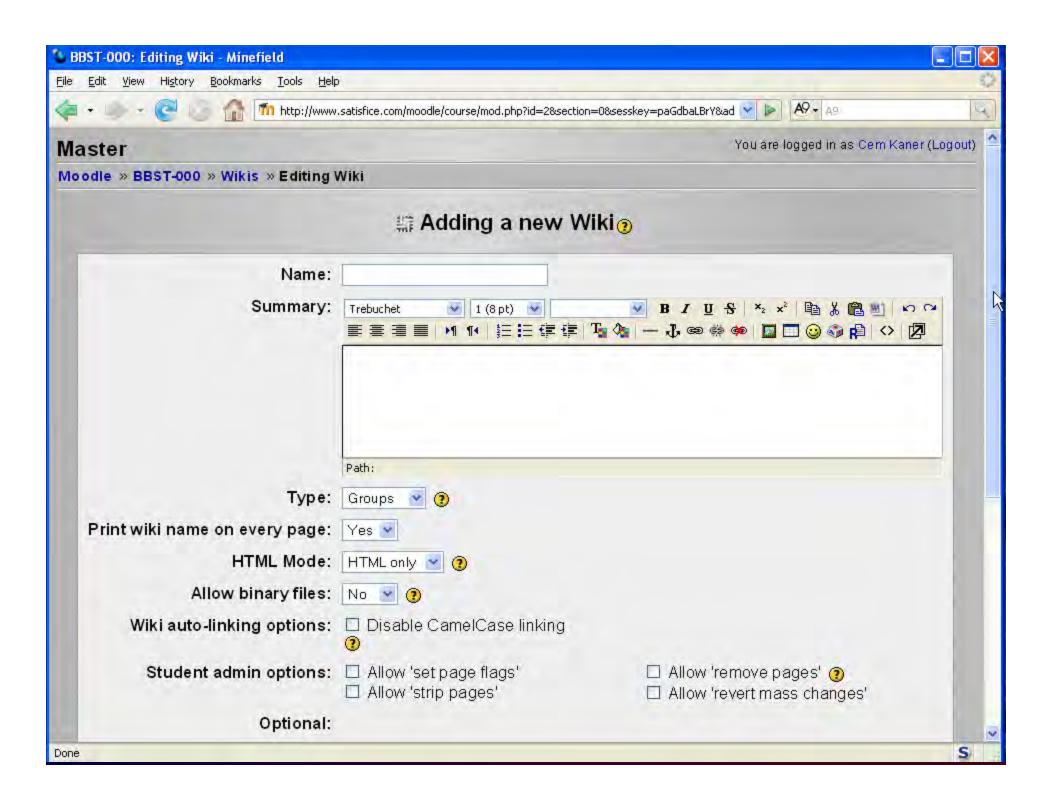










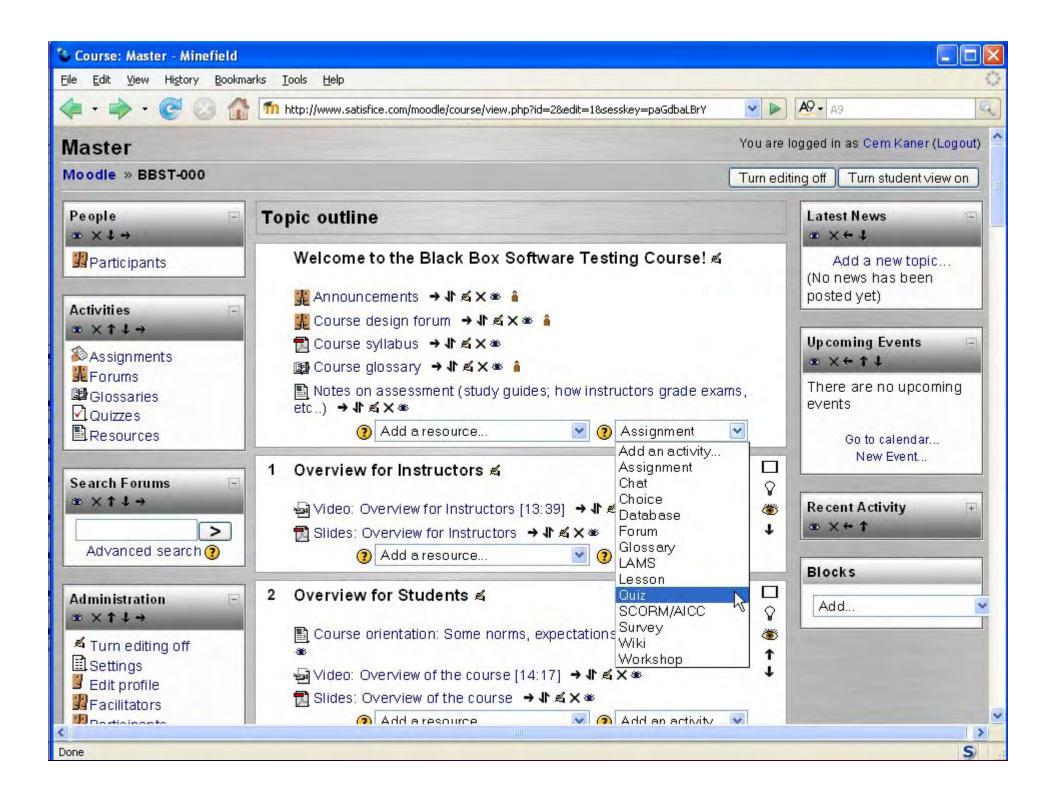


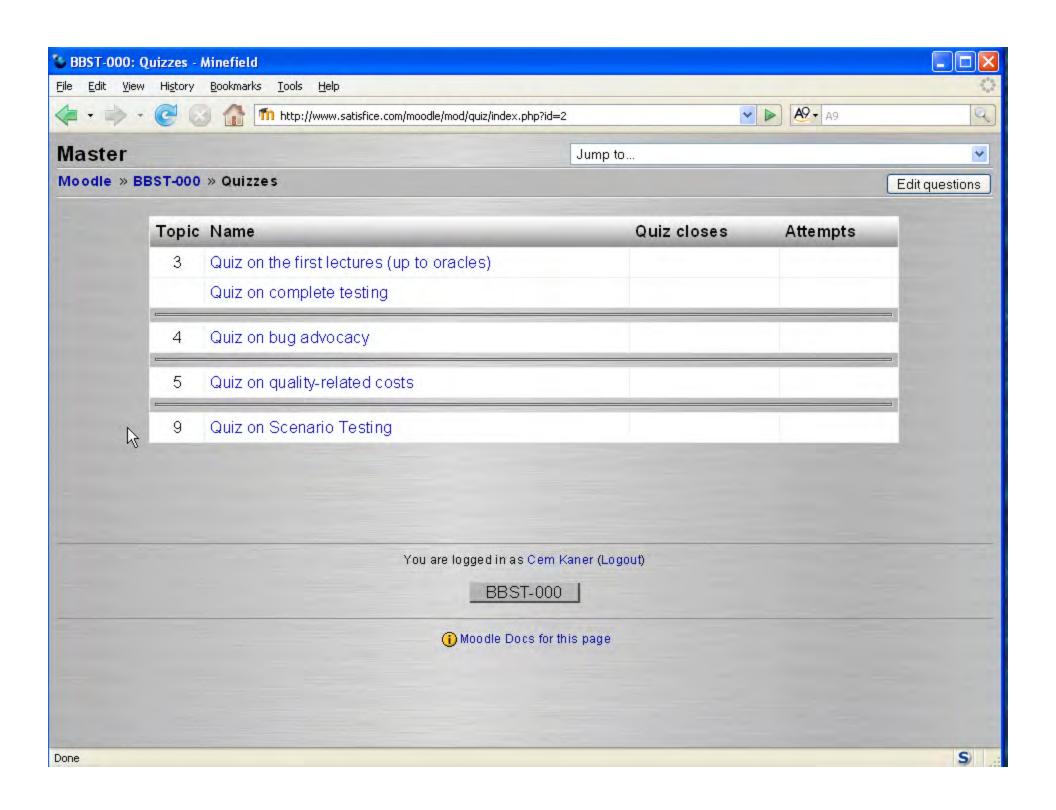


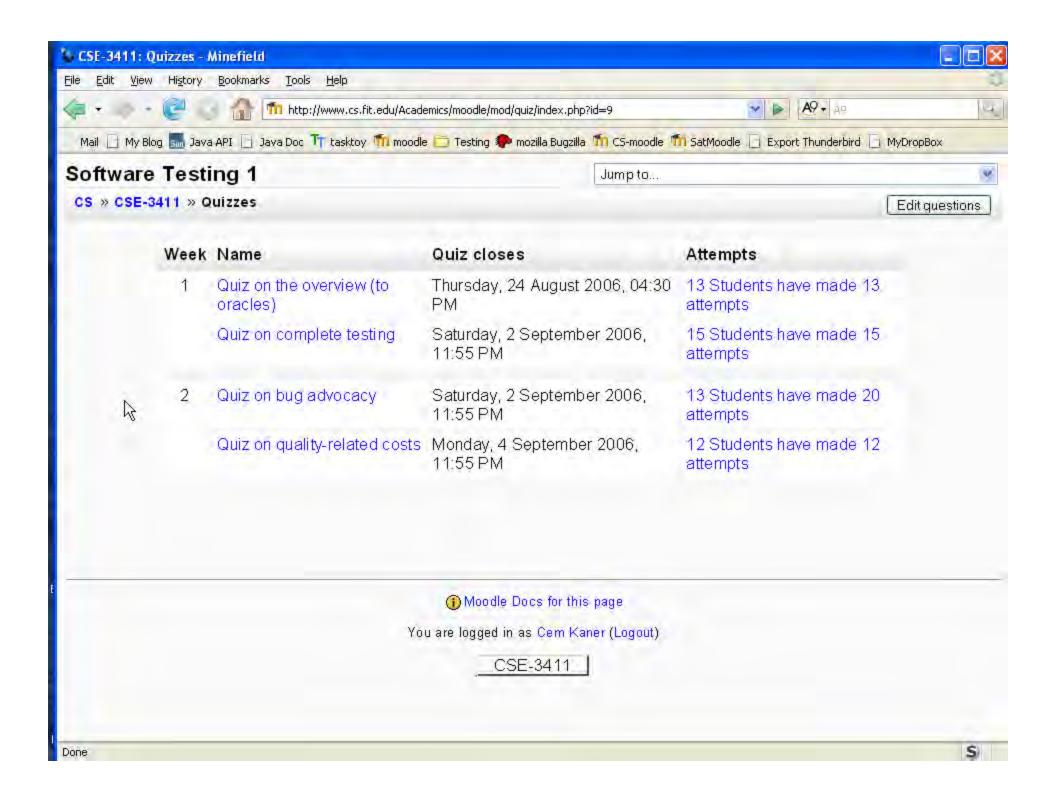
Stories for Developers

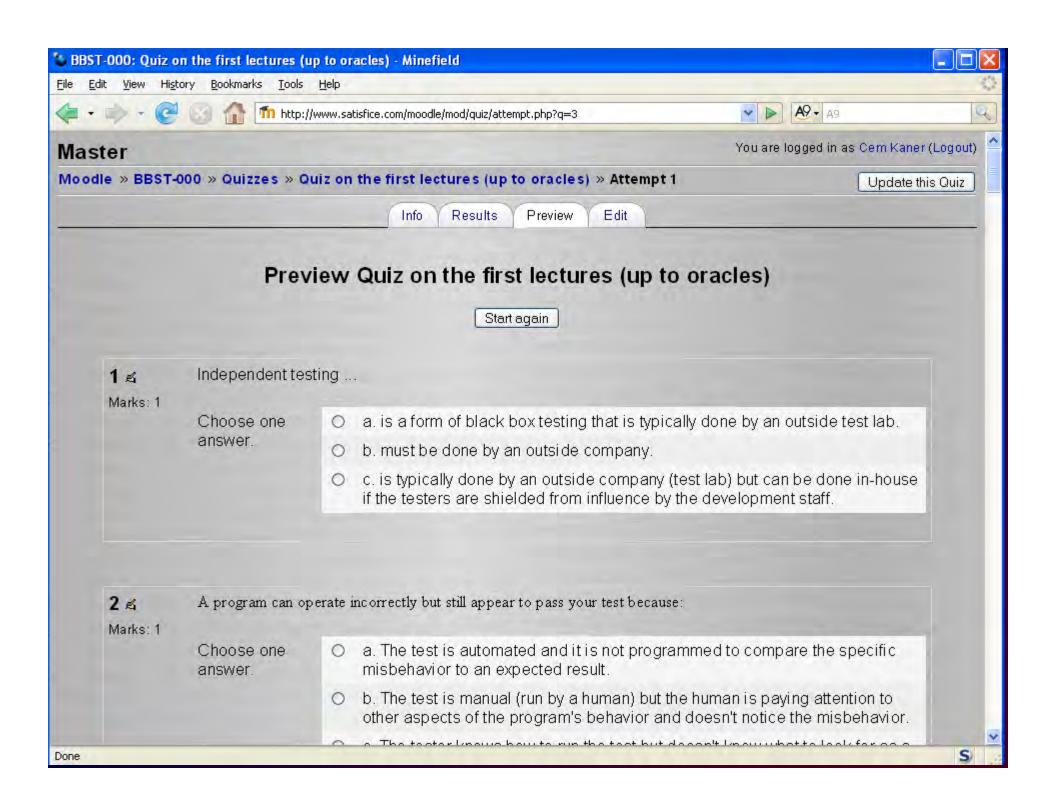
Examinee or Self-studier

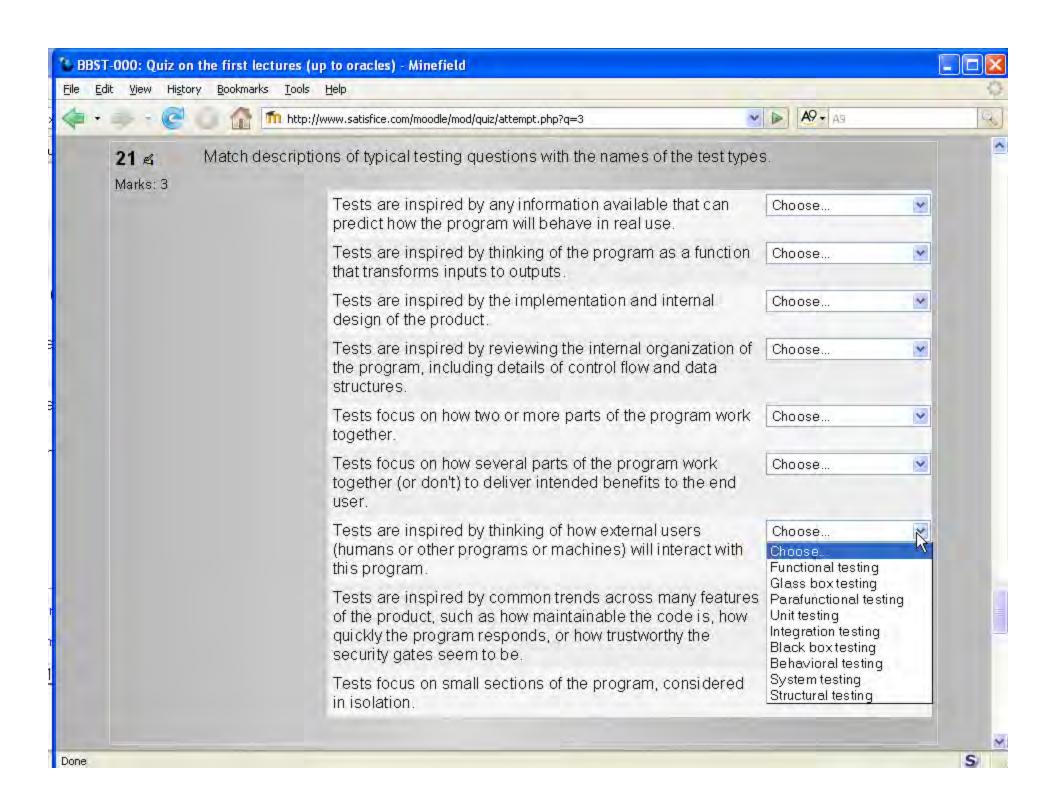
- I want to be able to exclude questions based on specific criteria (NOT anything from John Doe, NOT anything from gaming industry) -- Becky
- I want to be able to compare my performance to others. -- Becky
- . I want to be able to print a pretty certificate with my results on the test. -- Becky
- I want to be able to print (to printer or pdf) a copy of my results, including correct answers and comments on each question. --Becky
- I want to be able to see the question along with the answers and comments in a preview. --Becky
- I want to search for questions by topic or test technique. -- Becky
- I want to search for questions by style of question in combination with other criteria. For example, I want to find all true/false questions on domain testing by a specific author. --Becky
- I want to be able to use multiple key words. -- Becky
- I want to be able to extract the questions that pass through my filters in tab-delimited format. -- Becky
- Lawant to be able to deporate a practice feet according to enecific criteria (e.g. number of questions, enecific context

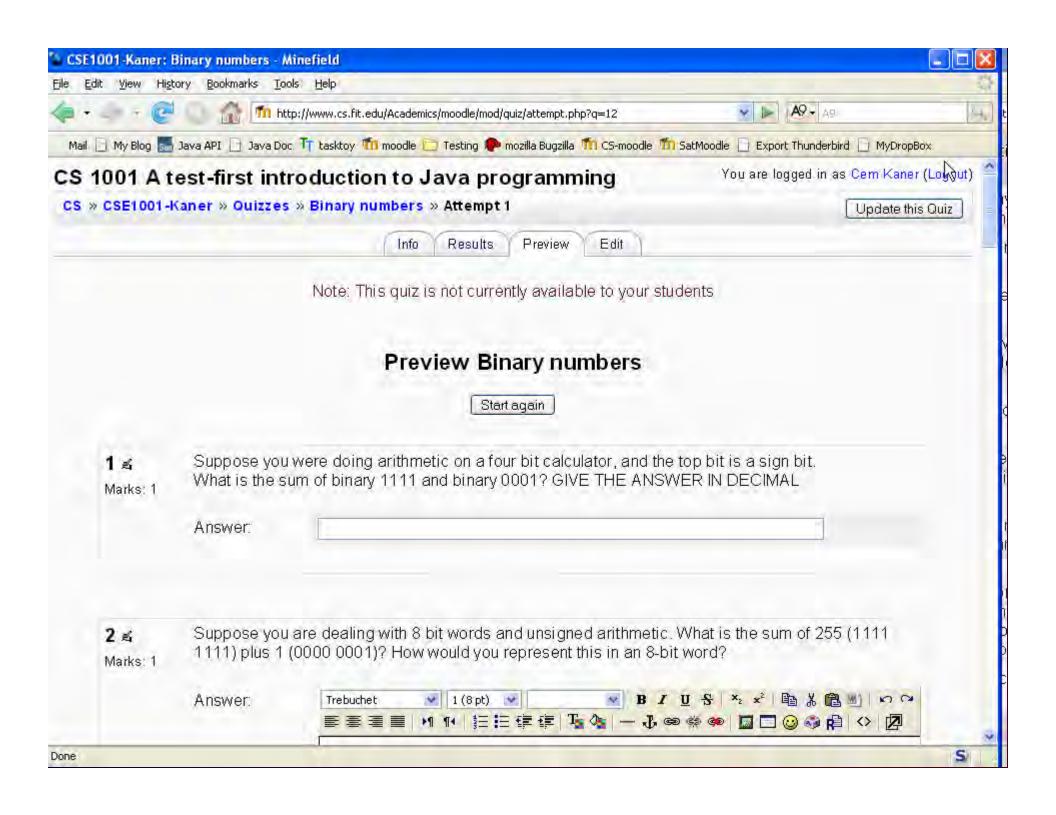


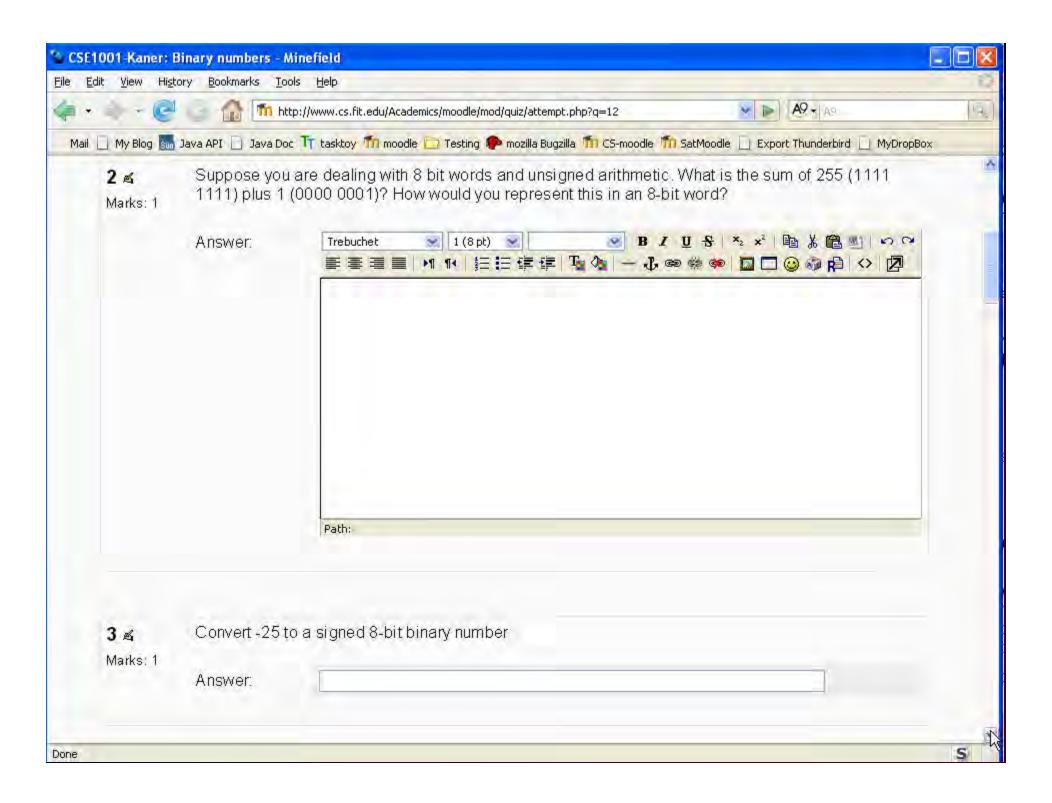


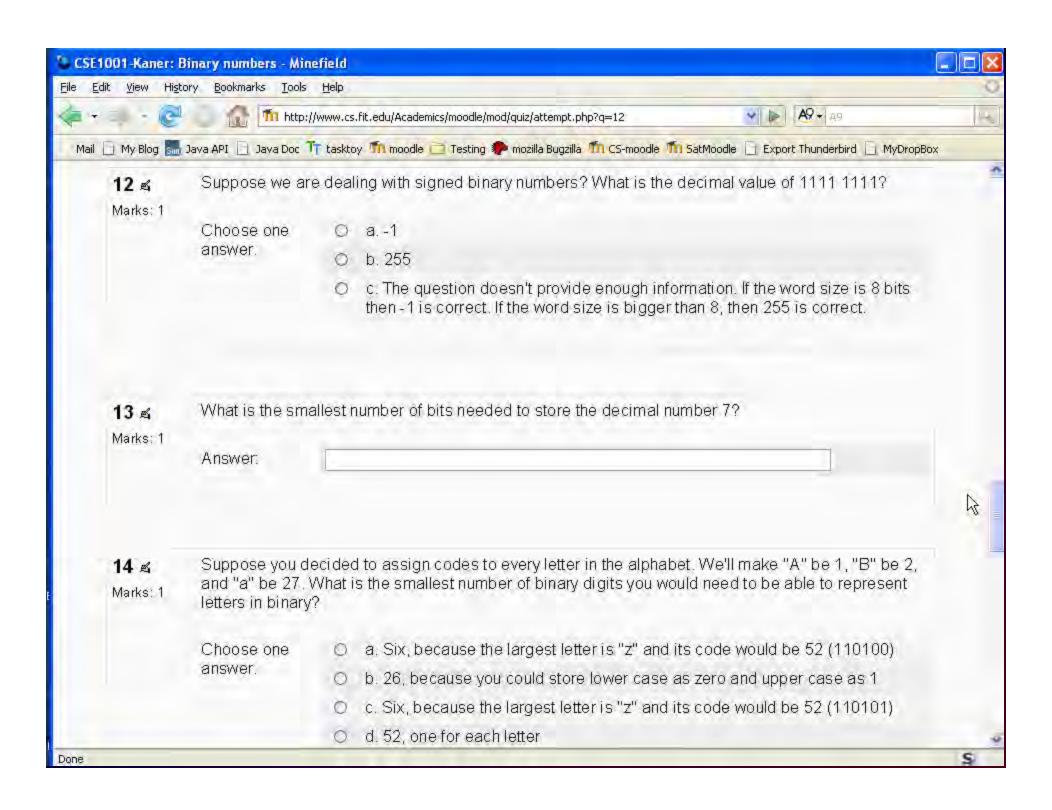


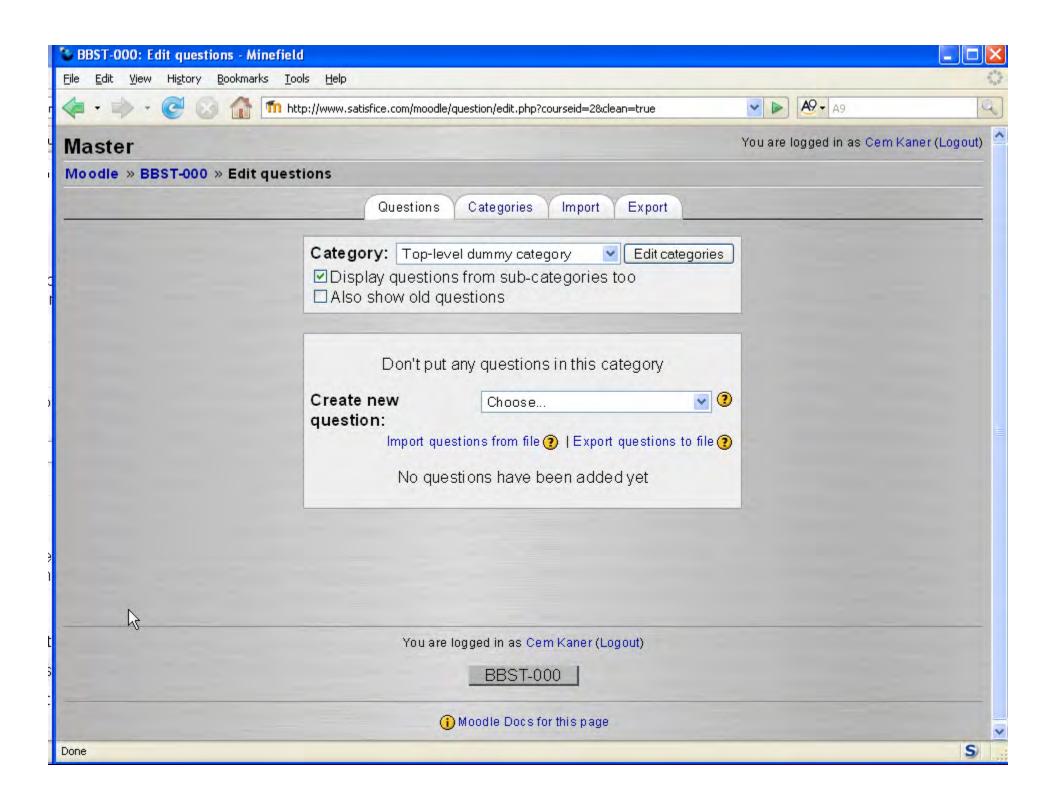


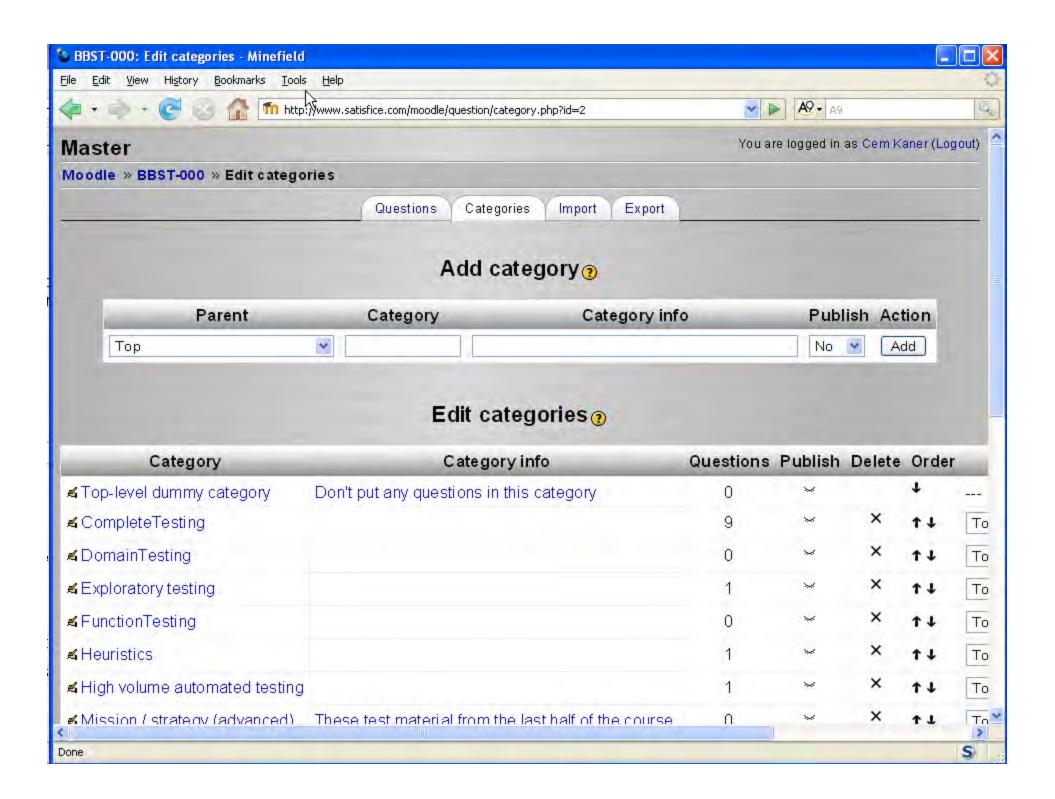


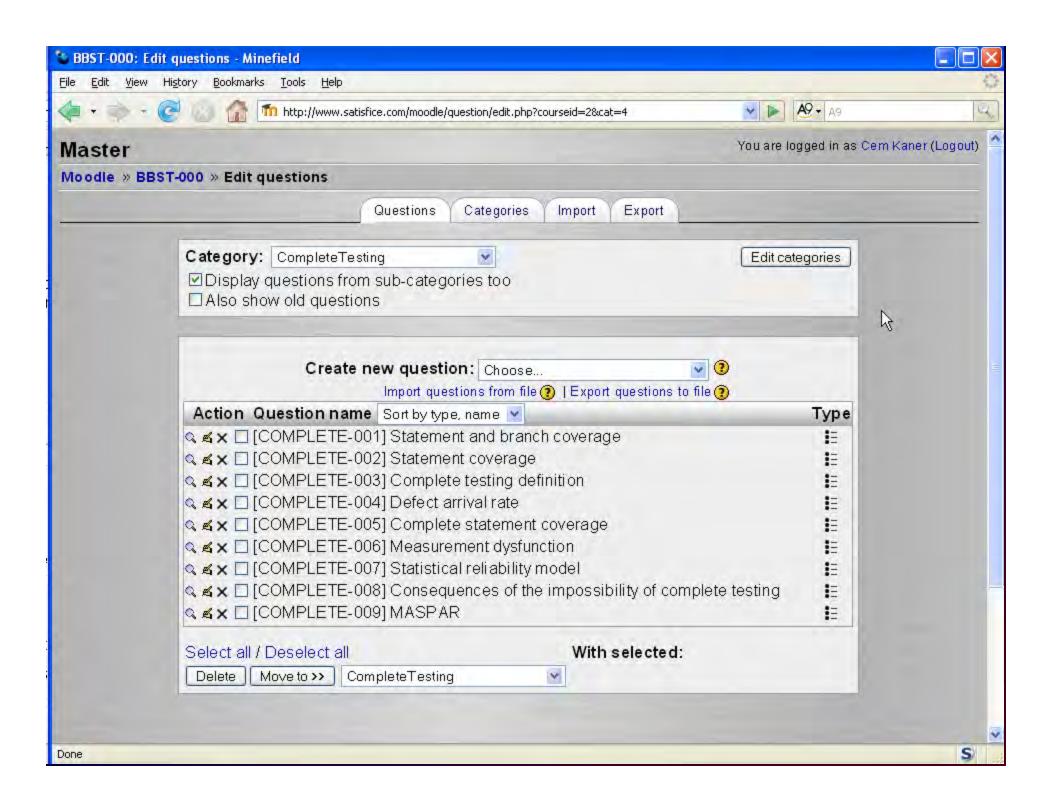


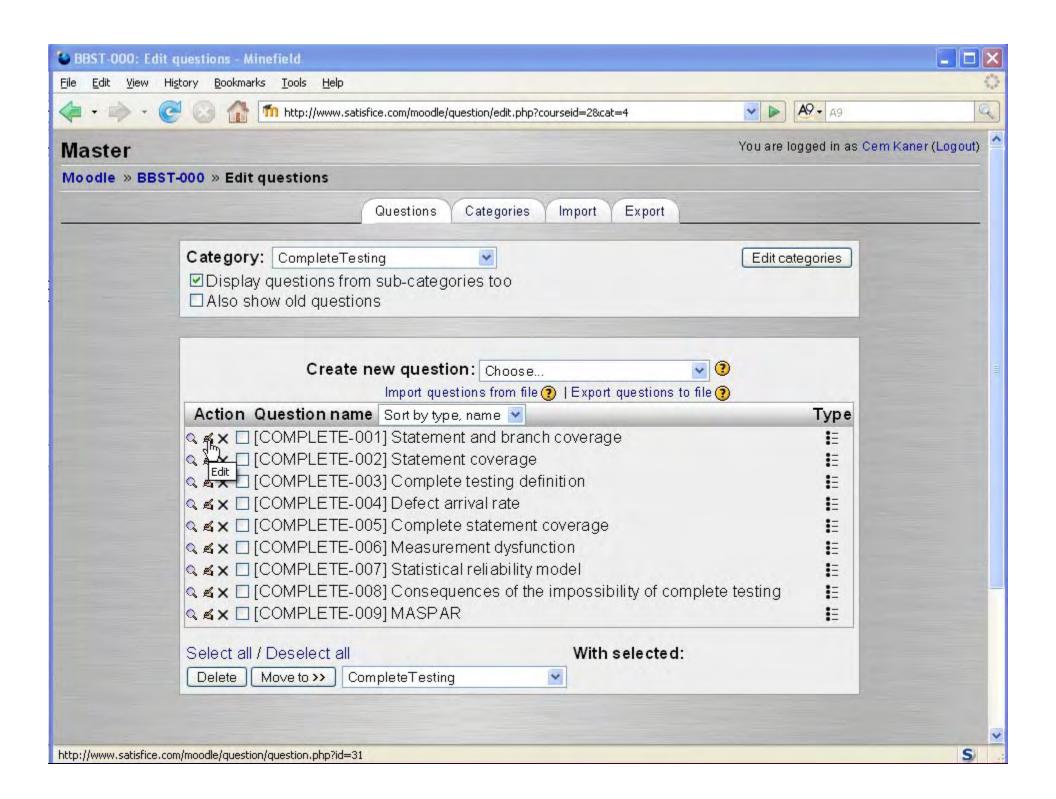


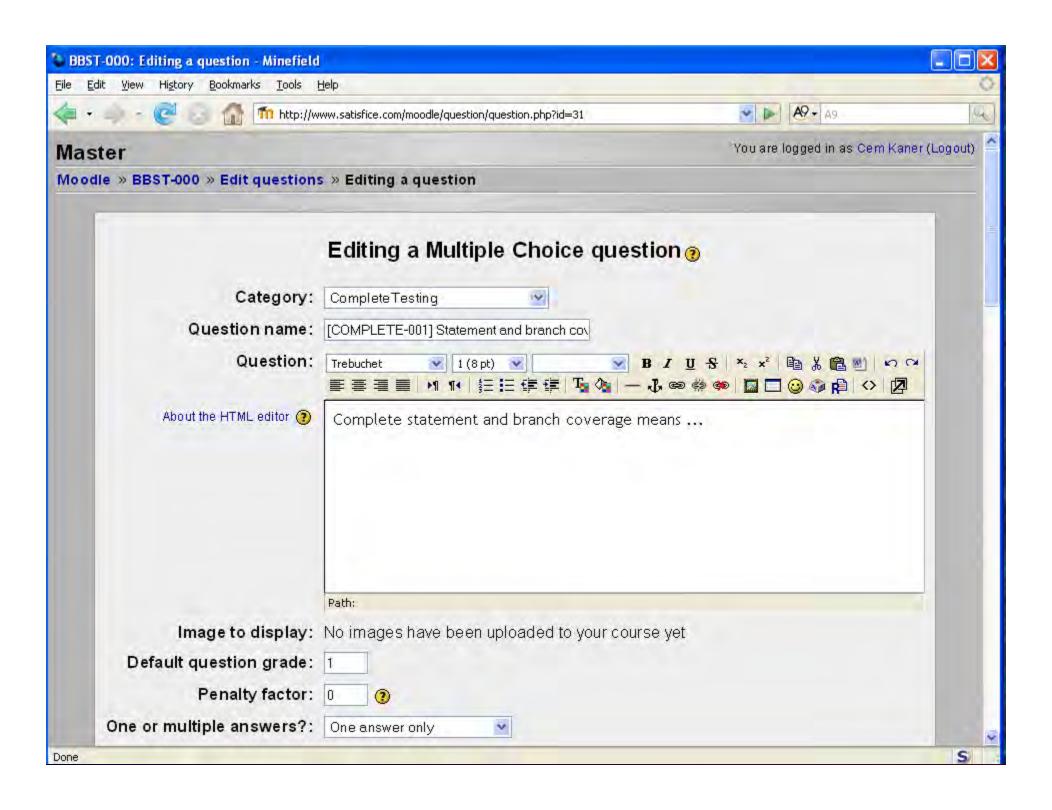


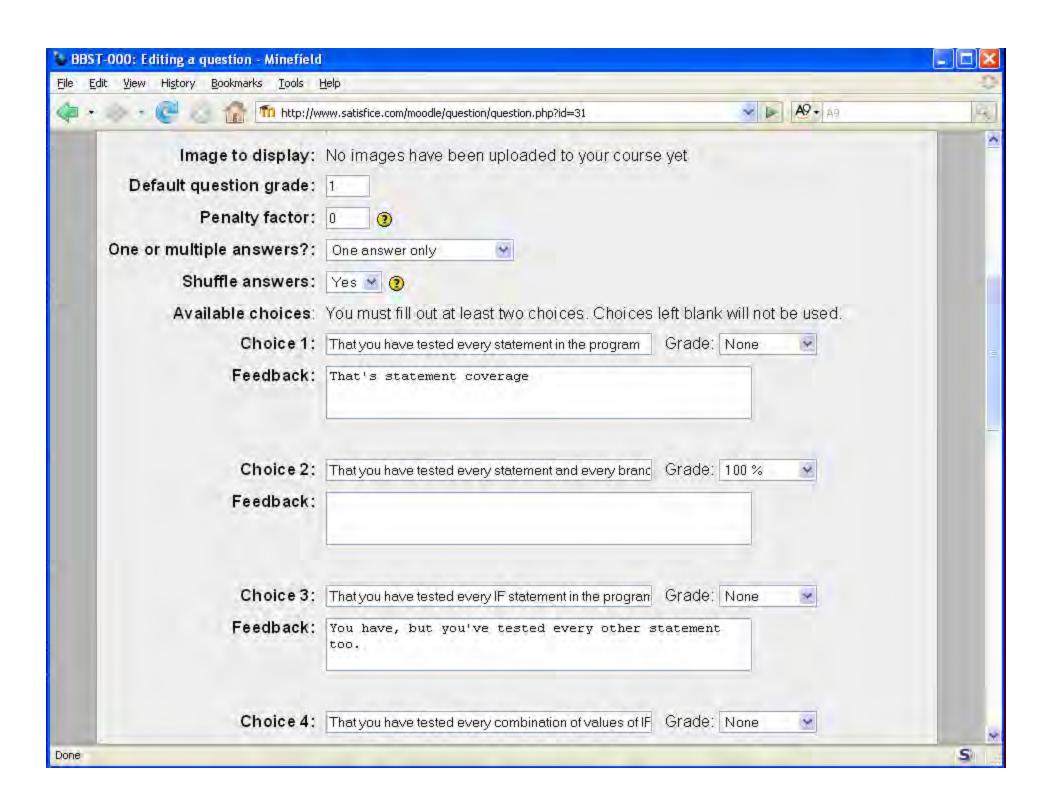


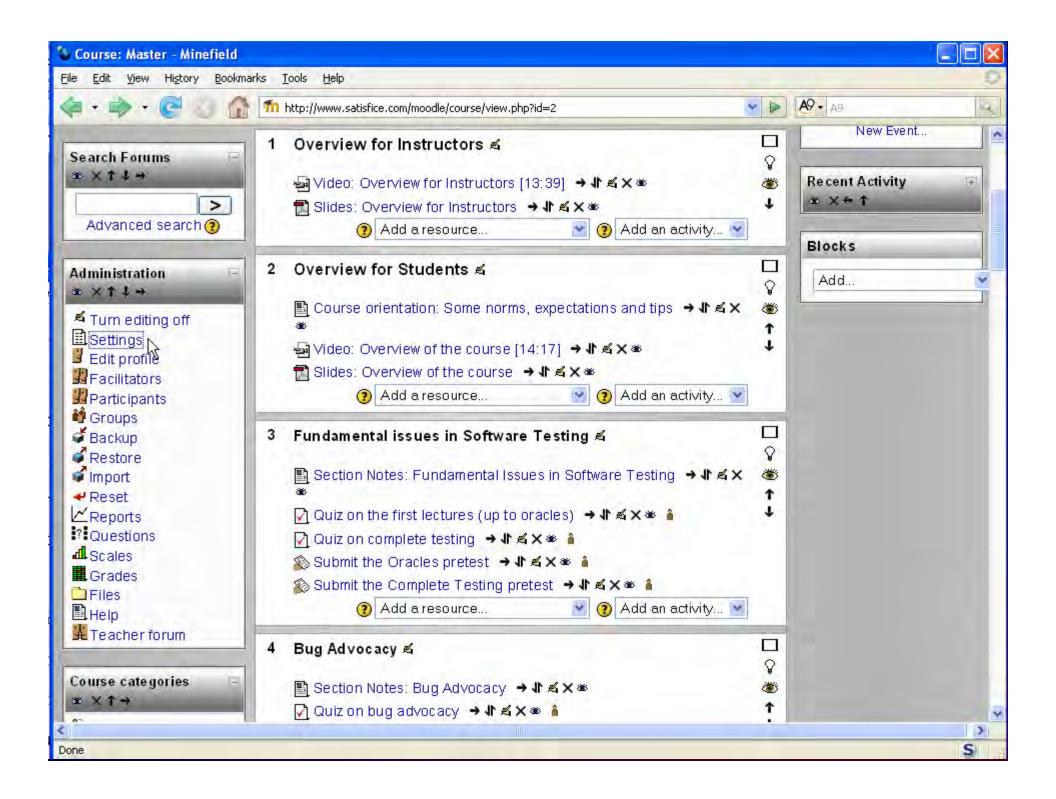


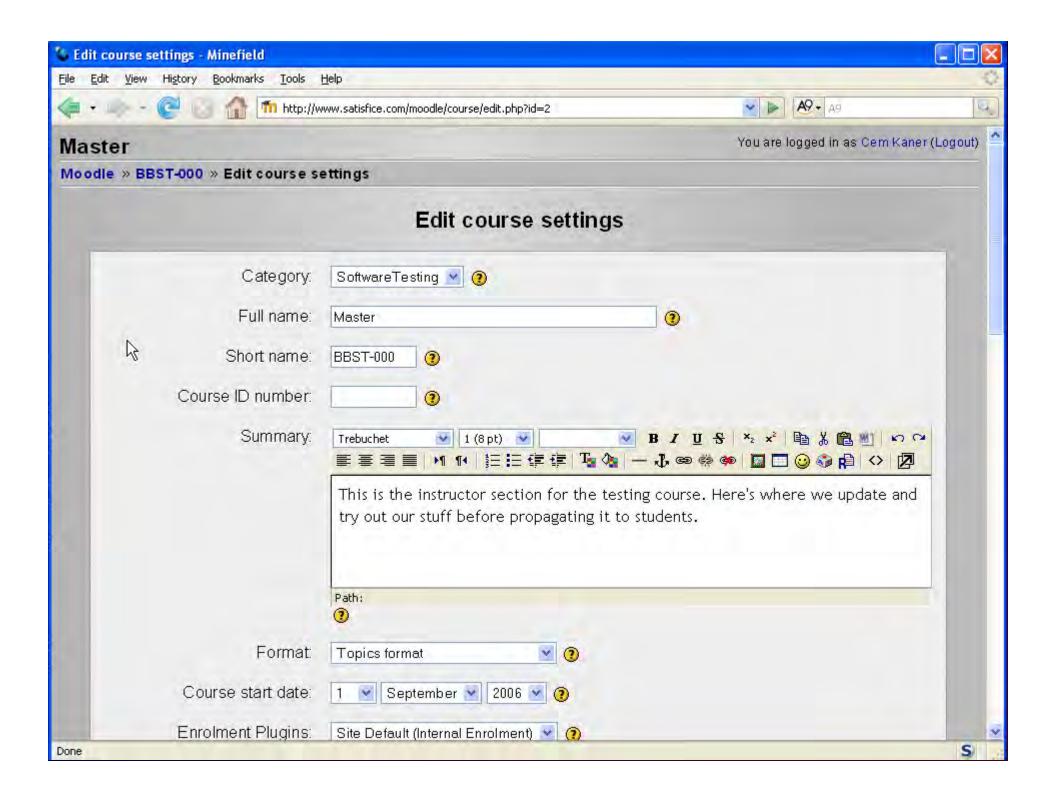


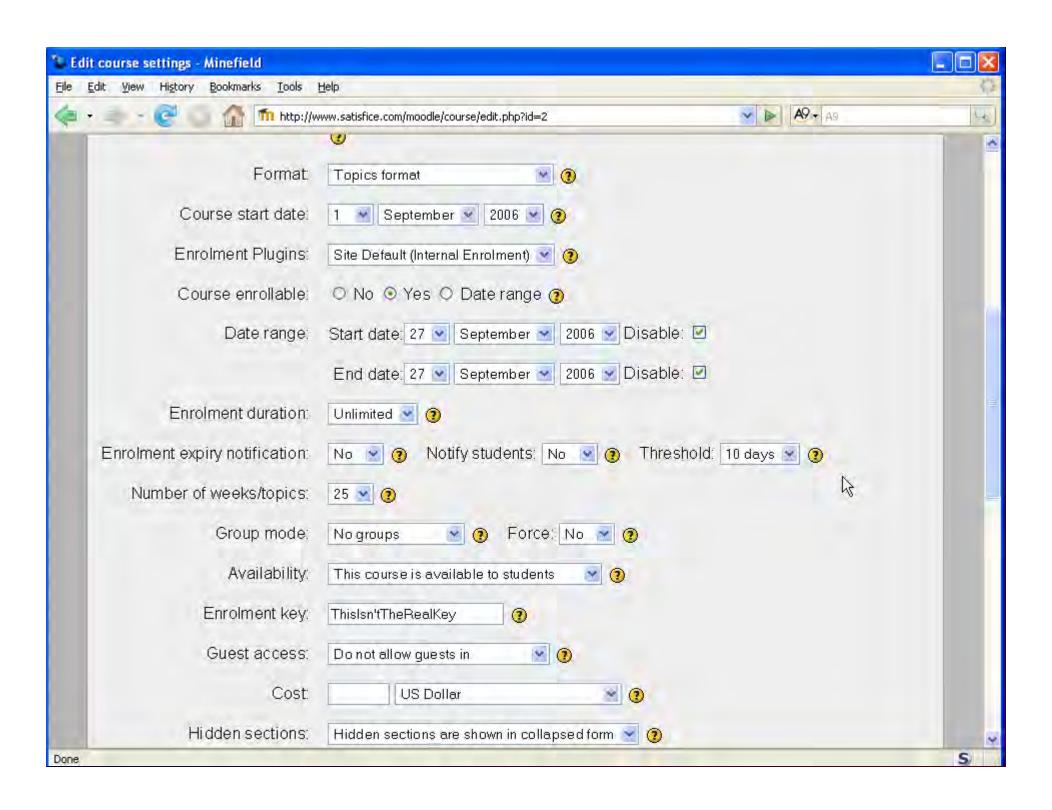






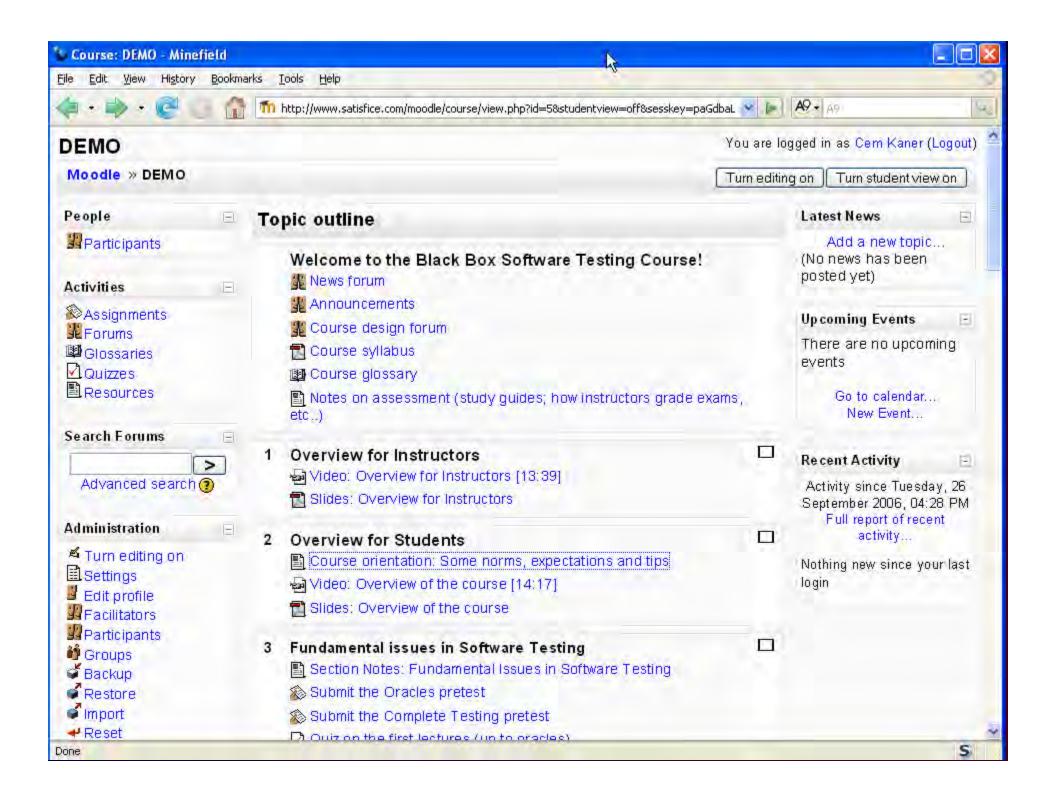


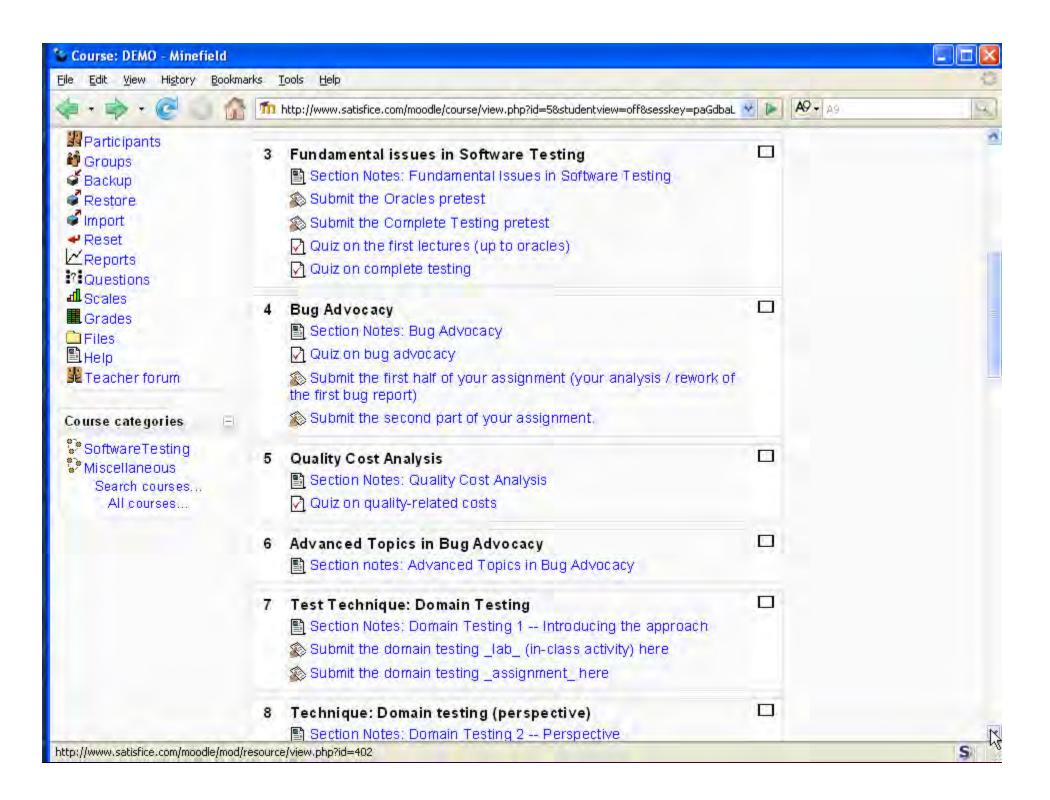


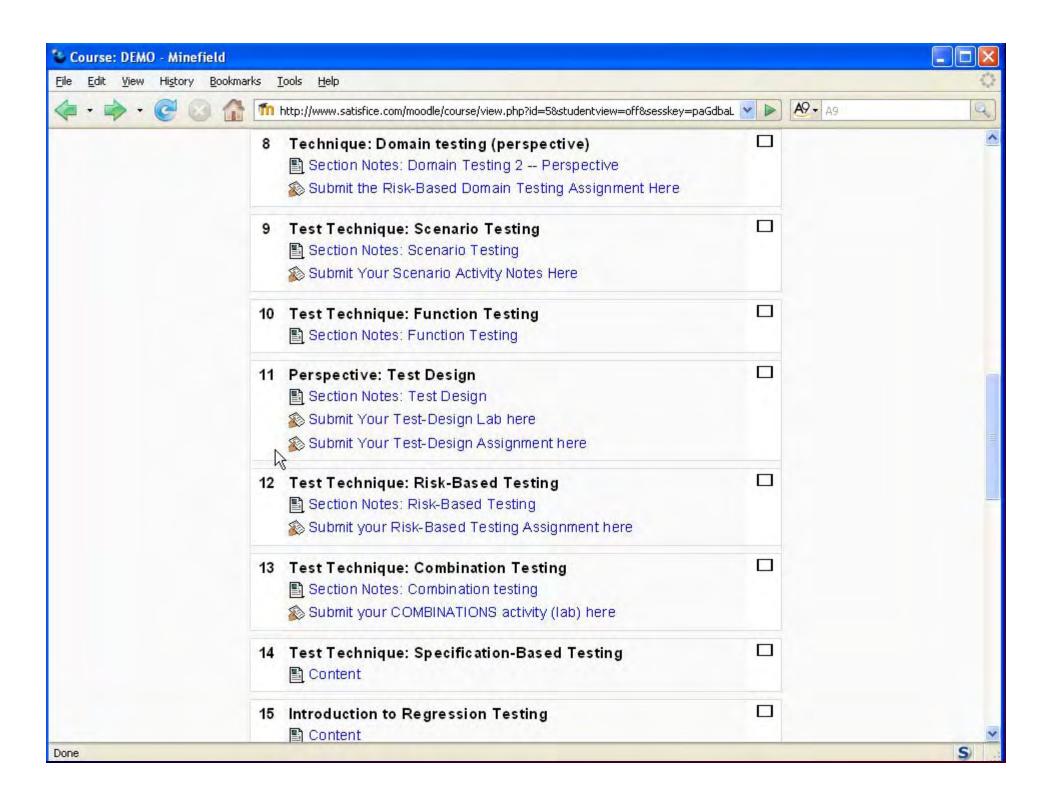


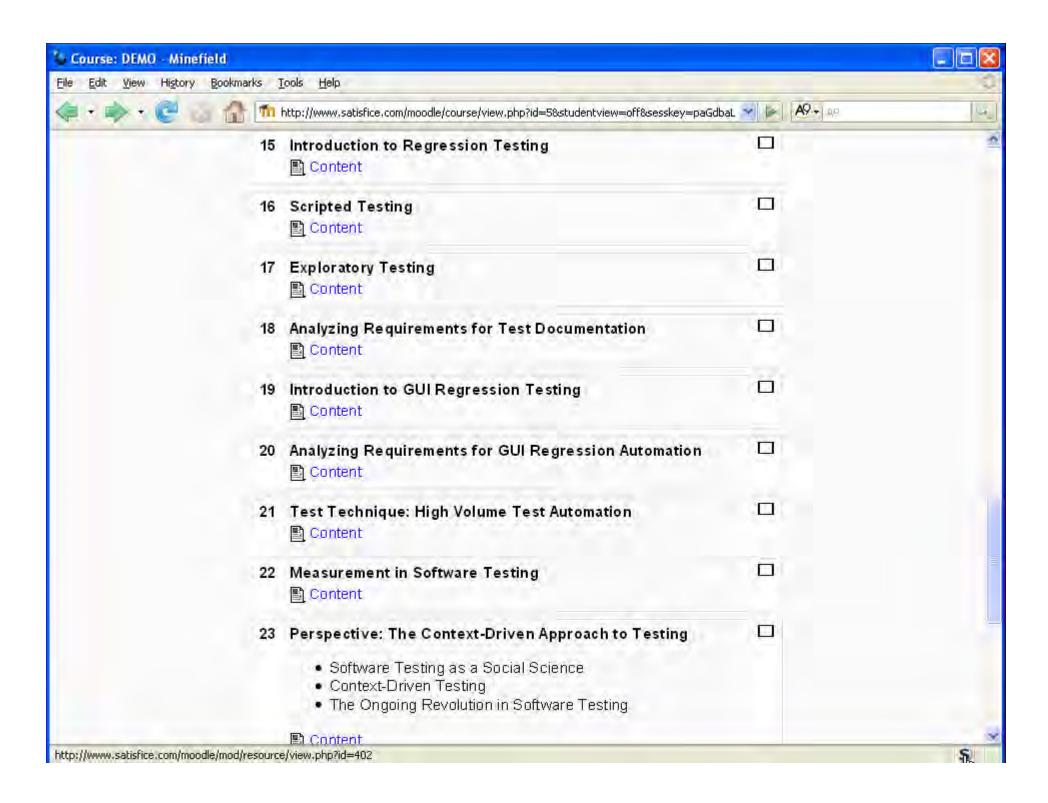
Overview

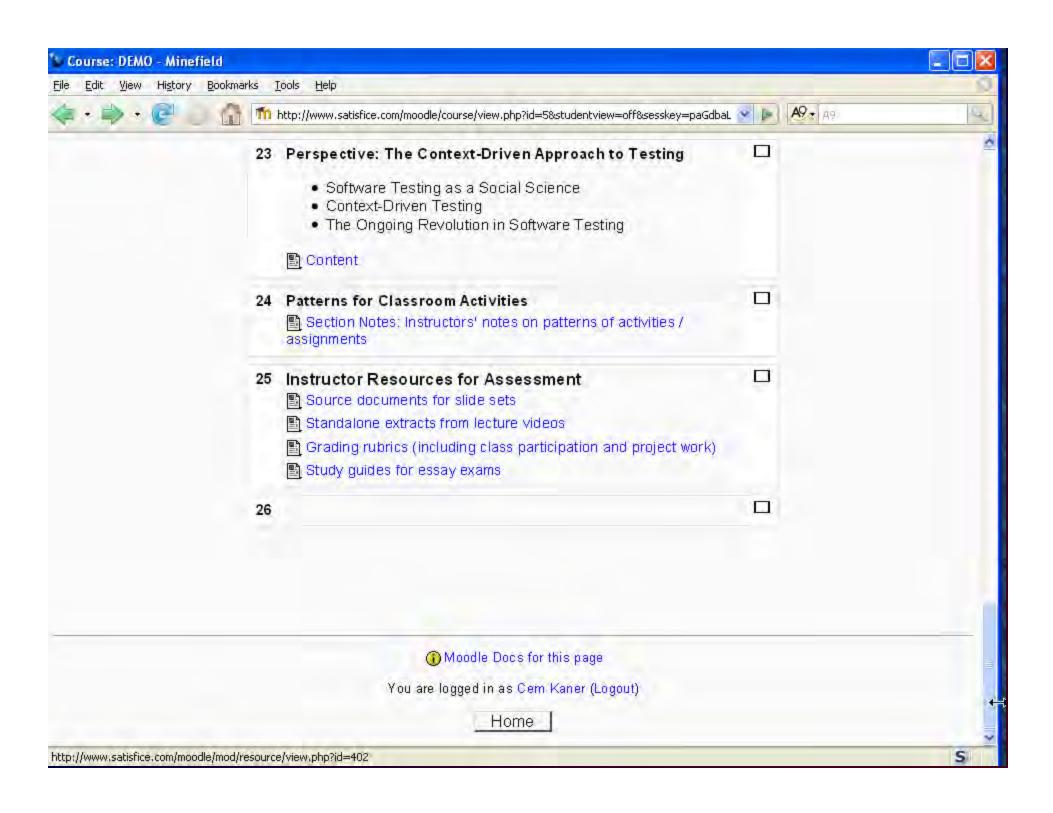
- Tour of the Moodle course management system
- Tour of the Black Box Software Testing Course on Moodle
- Overview of use of material like this in the workplace
- Dealing with the instructional challenges of a cognitively complex field of study

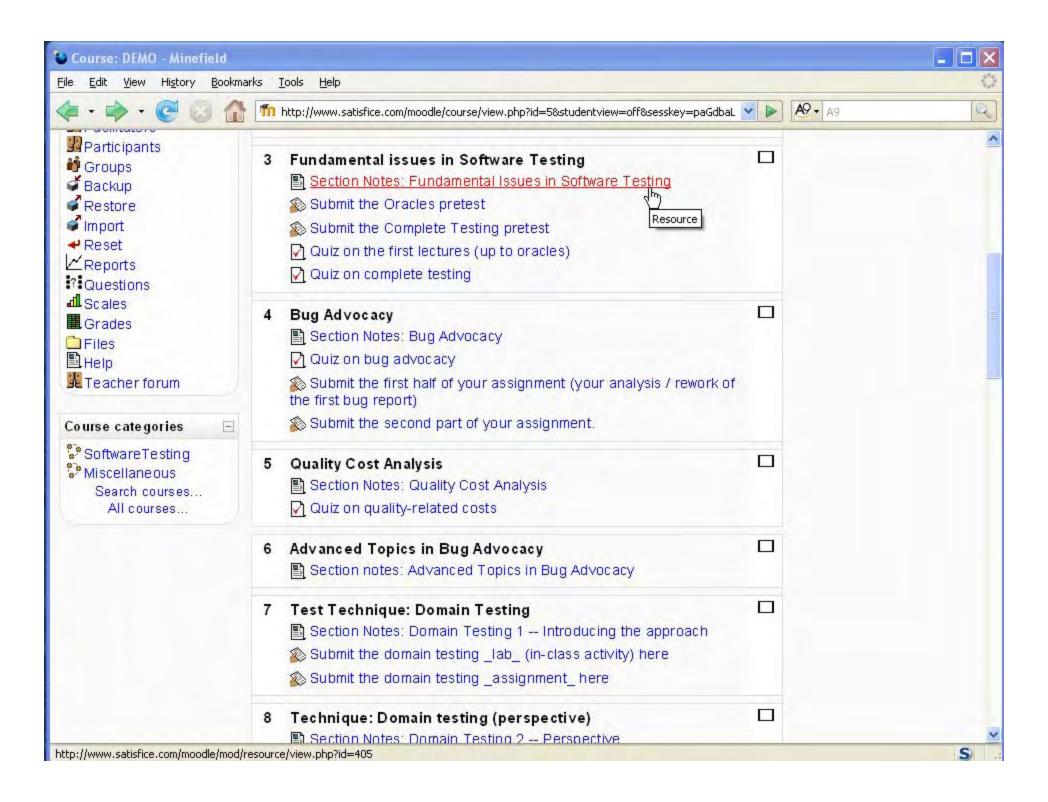














Fundamental Issues in Software Testing

Black box testing is the craft of testing a program from the external view. We look at how the program operates in its context, getting to know needs and reactions of the users, hardware and software platforms, and programs that communicate with it.

Testers who do primarily black box testing account for 20% to 60% of the technical staff on a typical software development project--there is an enormous market for skilled testers.

Testing is often misperceived as a fairly routine set of procedures for verifying that a program is correct (as if we could actually do that) or for finding bugs. In fact, skilled testing is a <u>cognitively complex activity</u>, as difficult and as creative as designing and writing code.

This opening section of the course looks at the variety of missions given to test groups and a few of the key problems that make testing so difficult and so interesting.

Reference Materials:

- Videos
 - Introducing the fundamental issues [5: 54] [SLIDES]
 - Mission and strategy of the testing effort [7:51] [SLIDES]
 - The oracle problem [19:07] [SLIDES]
 - The measurement problem and the impossibility of complete testing (Part 1) [29:37] [SLIDES (parts 1 & 2)]
 - The measurement problem and the impossibility of complete testing (Part 2) [26:44]

[On some browsers, clicking on a video link to play the video will not work. To play the video, download it to your disk and play the downloaded copy with Windows Media Player 9 or later.]

- Articles
 - o Hoffman: Heuristic test oracles

How the Course Works

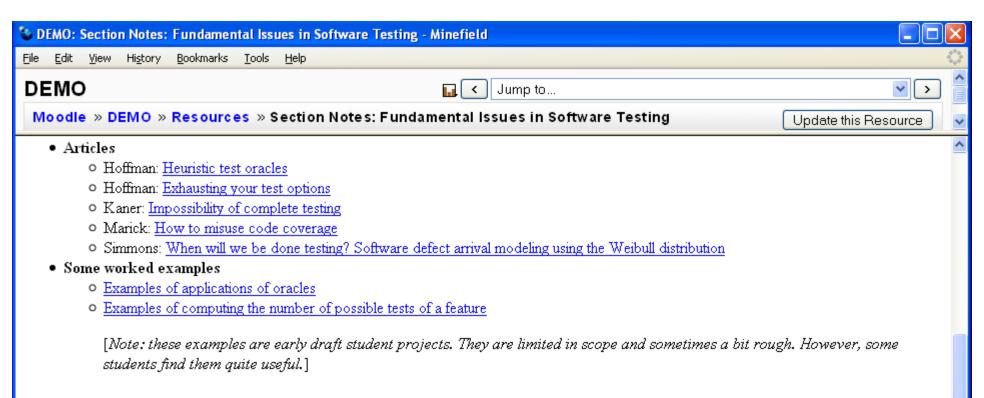
- Students watch the video before coming to class
- Students often work through an open-book quiz before coming to class
- We spend classroom time on
 - coached activities
 - facilitated discussions
 - group feedback (lecture) when I see a class-wide problem
- We apply the material in
 - in-class activities
 - out-of-class assignments

Lectures On-Line

- The results seem good
 - Good student satisfaction
 - Exam results aren't as different as I expected

Not enough time for the activities

In an in-house course, time is not constrained by the same type of schedule. It is constrained by value to the project and the staff.



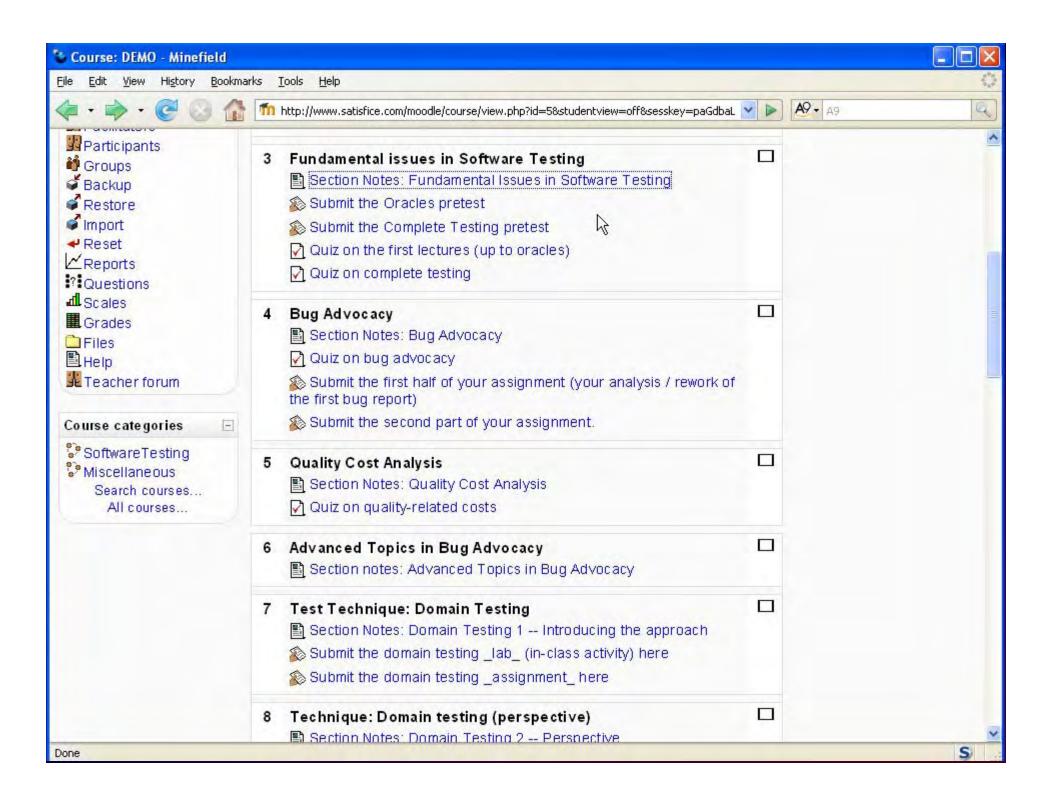
Activities and Assessments:

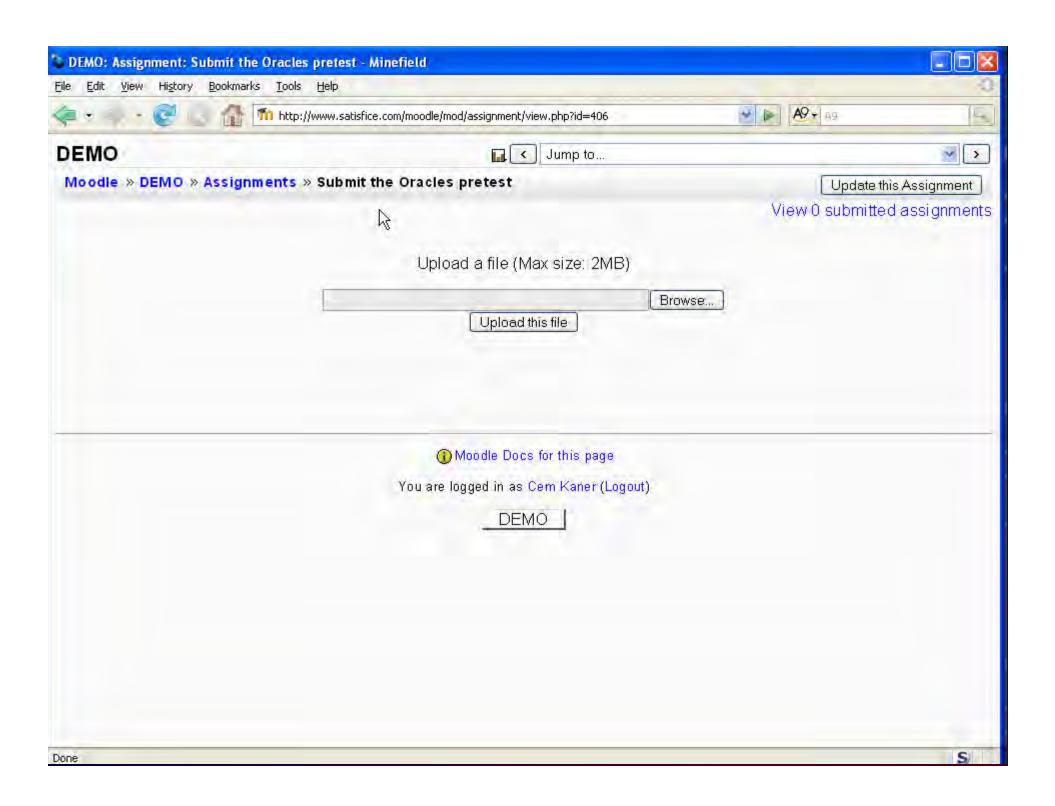
- Pre-test on oracles-- Submit your answer to this on the main moodle screen
- Pre-test on complete testing -- Submit your answer to this on the main moodle screen
- Review / drill questions -- These are available from the main moodle screen
- · Activity: Contrasting strategies for testing the same program.
- · Activity: Statement and path coverage of simple program fragments.

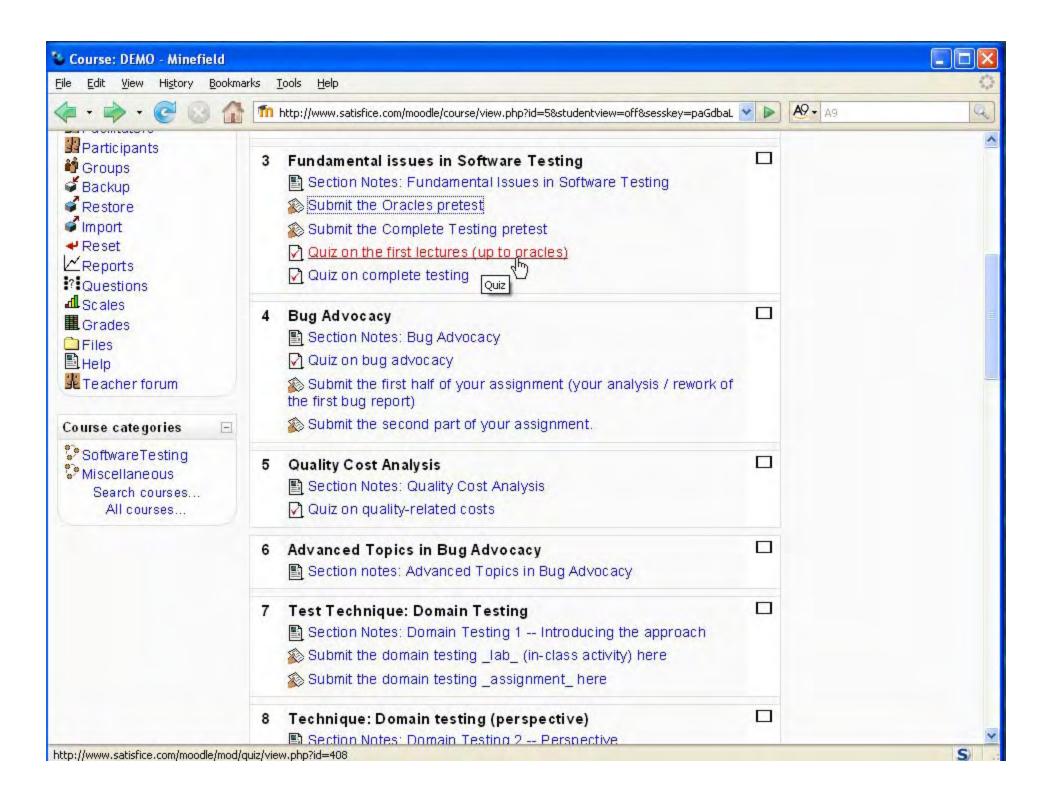
Summary of the Learning Unit

Software testing is an investigation conducted to provide quality-related information about a product. We test in many ways, looking for different types of information. We do the work on behalf of stakeholders (such as project managers) who need the information to improve the product or to make some decision such as whether to release the program for use or to sue the company that made the program for its provable defects.

We open our discussion of testing with a quick look at four key challenges:







Sample Activity: Contrasting Missions

- Your group is testing a spreadsheet / database. Please consider what your testing strategy should be and what types of test documentation to deliver.
- Different groups consider this question:
 - Traditional end-of-cycle test group
 - Development support near start of project
 - Testing a character database for a game
 - Testing a custom application for a medical device maker
- Groups report back, either by report/discussion to full group or by rotation of group representatives into discussion groups

Application Under Test

- We pick a well-known product
- Students apply what they learn to that product
- Typically, I use an open source product because it avoids NDA problems, students can show their work at interviews
- Facilitates student learning (application level and above)
- Facilitates student transfer of skills / knowledge to the workplace

In an in-house course, the AUT is your product

Study Guides

- www.testingeducation.org/ko4/BBSTreviewfall2005.htm
- 100 questions, include all candidates for mid-term and final exam
- Students prepare answers together, assess each other's work
- I can require well-organized, thoughtful answers
- Fosters strategic preparation
- Reduces disadvantage of students whose native language is not English
- Creates cooperative learning tasks that should help limited-English-proficiency students improve language skills

Study Guides

- Study guide results
 - Students inexperienced with these, often blow the first test
 - Make-up mid-terms
 - Replace grade, not average, not best 1 of 2 results
 - Students who take it improve more (1st test compared to final exam) than students who did not take it
 - Practice effect, motivation confound
 - Writing is better, answers are better, I have greater freedom to grade less forgivingly
 - Many students told me this was the most valuable learning experience in the course, and the most time-consuming

Study Guides

- In-house use:
 - Focus discussion of course materials
 - Potential interview questions, especially if you revise them to apply to your class of product

Assessing student reaction

- Chose the Student Assessment of Learning Gains http://www.flaguide.org/cat/salg/
- Measures student perceptions of their 'gains' in learning
- Customizable
- Administered online
- FREE
- Beats the standard course evaluation form!
- Students each spent over an hour providing their evaluation.

Overview

- Tour of the Moodle course management system
- Tour of the Black Box Software Testing Course on Moodle
- •Overview of use of material like this in the workplace
- Dealing with the instructional challenges of a cognitively complex field of study

Instruction in the Workplace

- The opportunity:
 - Build on the strengths of commercial instruction
 - Avoid the weaknesses of academic instruction

Commercial vs. Academic

- Drive-by teaching
 - 2-5 days, rapid-fire ideas,
 visiting instructor
- Broad, shallow coverage
- Time constraints limit activities
- No time for homework
- No exams
- Coached, repeated practice seen as time-wasting
- Familiarity
- Work experience helps to bring home concepts
- Richer grounding in real practice
- Some (occasional) student groups share a genuine current need
- Objective: one applicable new idea per day

- Local teaching
 - Several months, a few hours per week, students get to know instructor
- Deeper coverage
- Activities expected to develop skills
- Extensive homework
- Assessment expected
- Coached, repeated practice is highly appreciated
- Capability
- Students have no work experience, need context
- Harder to connect to real practice
- Students don't naturally come to a course as a group with a shared problem
- Expect mastery of several concepts and skills

Build on the Strengths

- Adopt a deliberate, slow pace
- Work as a learning team
- Focus on application of the tasks to current projects, use the course as a vehicle for tinkering with productivity and creativity of your day-to-day work

Demonstrating current value of the learning experience builds management support for continued investment

One vision of the in-house course

- Meet weekly for a year
- Watch 10-25 minutes of video in advance
- Discuss the lesson and its applicability
- Over the next week, try to apply it on the job to the current project(s) in test
- Discuss the application results in the next week or move to the next segment.
- At the end of the course, students know how things fit into their environment, and have multiple examples (from multiple student colleagues).

Overview

- Tour of the Moodle course management system
- Tour of the Black Box Software Testing Course on Moodle
- Overview of use of material like this in the workplace
- •Dealing with the instructional challenges of a cognitively complex field of study

The instructional challenge, as I see it

Software testing

is cognitively complex,

requires critical thinking,

effective communication, and

rapid self-directed learning.

Software testing is a process of empirical, technical investigation of the product under test conducted to provide stakeholders with quality-related information.

What's a test technique? Ten dominating techniques

- Function testing
- Specification-based testing
- Domain testing
- Risk-based testing
- Scenario testing
- Regression testing
- Stress testing
- User testing
- State-model based testing
- High volume automated testing

These are
10 common
Examples.

There are many
Others.

Black Box Software Testing

Copyright @ 2003

Cem Kaner & James Bach

http://www.testingeducation.org/BBST/BBST--IntroductiontoTestDesign.html

Test attributes

To different degrees, good tests have these attributes:

- · Power. When a problem exists, the test will reveal it.
- Valid. When the test reveals a problem, it is a genuine problem.
- Value. It reveals things your clients want to know about the product or project.
- Credible. Your client will believe that people will do the things that are done in this test.
- Representative of events most likely to be encountered by the user. (xref. Musa's Software Reliability Engineering).
- Non-redundant. This test represents a larger group that address the same risk.
- Motivating. Your client will want to fix the problem exposed by this test.
- · Performable. It can be performed as designed.
- Maintainable. Easy to revise in the face of product changes.
- Repeatable. It is easy and inexpensive to reuse the test.
- Pop. (short for Karl Popper) It reveal things about our basic or critical assumptions.
- Coverage. It exercises the product in a way that isn't already taken care of by other tests.
- Easy to evaluate.
- · Supports troubleshooting. Provides useful information for the debugging programmer.
- Appropriately complex. As the program gets more stable, you can hit it with more complex tests
 and more closely simulate use by experienced users.
- · Accountable. You can explain, justify, and prove you ran it.
- Cost. This includes time and effort, as well as direct costs.
- Opportunity Cost. Developing and performing this test prevents you from doing other work

Black Box Software Testing

Copyright © 2003

Cem Kaner & James Bach

25

Contexts Vary Across Projects

Testers must learn, for each new product:

- What are the goals and quality criteria for the project
- What skills and resources are available to the project
- What is in the product
- How it could fail
- What the consequences of potential failures could be
- Who might care about which consequence of what failure
- How to trigger a fault that generates the failure we're seeking
- How to recognize failure
- How to decide what result variables to pay attention to
- How to decide what other result variables to pay attention to in the event of intermittent failure
- How to troubleshoot and simplify a failure, so as to better
 - (a) motivate a stakeholder who might advocate for a fix
 - (b) enable a fixer to identify and stomp the bug more quickly
- How to expose, and who to expose to, undelivered benefits, unsatisfied implications, traps, and missed opportunities.

It's kind of like CSI



MANY tools, procedures, sources of evidence.

- Tools and procedures don't define an investigation or its goals.
- There is too much evidence to test. tools are often expensive, so investigators must exercise judgment.
- The investigator must pick what to study, and how, in order to reveal the most needed information.

FPISODES

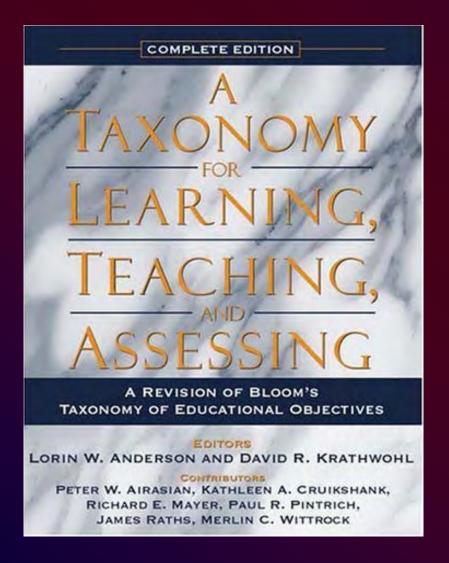
Missed an episode? Catch up on previous stories now.



moment to the max. Sex, drugs and a "High Roller Suite" are put at his disposal as he enters the ...more

Be among the first to know. Subscribe now for email updates

Characterizing Cognitive Complexity



Anderson & Krathwohl
 (2001) provide a modern
 update to Bloom's (1956)
 taxonomy

Characterizing Cognitive Complexity

		Cognitive Process Dimension					
Knowledge Dimension		Remember	Understand	Apply	Analyze	Evaluate	Create
	Factual	lecture	lecture				
	Conceptual	lecture	lecture				
	Procedural	lecture	lecture				
	Meta- Cognitive						

Anderson & Krathwohl, 2001

A Slight Variation for Testing

- Facts
- Concepts
- Procedures
- Cognitive strategies
- Models
- Skills
- Attitudes
- Metacognition
- The Testing Learning Concepts Taxonomy, Kaner & Bach, unpublished beta version

Variation for Testing: Facts

- A "statement of fact" is a statement that can be unambiguously proved true or false. For example, "James Bach was born in 1623" is a statement of fact. (But not true, for the James Bach we know and love.) A fact is the subject of a true statement of fact.
- Facts include such things as:
 - Tidbits about famous people
 - Famous examples (the example might also be relevant to a concept, procedure, skill or attitude)
 - Items of knowledge about devices (for example, a description of an interoperability problem between two devices)

Variation for Testing: Concepts

- A concept is a general idea. "Concepts are abstract in that they omit the differences of things in their extension, treating them as if they were identical." (wikipedia: Concept).
- In practical terms, we treat the following kinds of things as "concepts" in this taxonomy:
 - definitions
 - descriptions of relationships between things
 - descriptions of contrasts between things
 - description of the idea underlying a practice, process, task, heuristic (whatever)
- Here's a distinction that you might find useful.
 - Consider the oracle heuristic, "Compare the behavior of this program with a respected competitor and report a bug if this program's behavior seems inconsistent with and possibly worse than the competitor's."
 - If I am merely describing the heuristic, I am giving you a concept.
 - If I tell you to make a decision based on this heuristic, I am giving you a
- Sometimes, a rule is a concept.
 - A rule is an imperative ("Stop at a red light") or a causal relationship ("Two plus two yields four") or a statement of a norm ("Don't wear undershorts outside of your pants at formal meetings").
 - The description / definition of the rule is the concept
 - Applying the rule in a straightforward way is application of a concept
 - The decision to puzzle through the value or applicability of a rule is in the realm of cognitive strategies.
 - The description of a rule in a formalized way is probably a model.

Variation for Testing: Procedures

- "Procedures" are algorithms. They include a reproducible set of steps for achieving a goal.
- Consider the task of reporting a bug. Imagine that someone has
 - broken this task down into subtasks (simplify the steps, look for more general conditions, write a short descriptive summary, etc.)
 - and presented the tasks in a sequential order.
- This description is intended as a procedure if the author expects you to do all of the steps in exactly this order every time.
- This description is a cognitive strategy if it is meant to provide a set of ideas to help you think through what you have to do for a given bug, with the understanding that you may do different things in different orders each time, but find this a useful reference point as you go.

Variation for Testing: Cognitive Strategies

"Cognitive strategies are guiding procedures that students can use to help them complete less-structured tasks such as those in reading comprehension and writing. The concept of cognitive strategies and the research on cognitive strategies represent the third important advance in instruction.

There are some academic tasks that are "well-structured." These tasks can be broken down into a fixed sequence of subtasks and steps that consistently lead to the same goal. The steps are concrete and visible. There is a specific, predictable algor ithm that can be followed, one that enables students to obtain the same result each time they perform the algorithmic operations. These well-structured tasks are taught by teaching each step of the algorithm to students. The results of the research on tea cher effects are particularly relevant in helping us learn how teach students algorithms they can use to complete well-structured tasks.

In contrast, reading comprehension, writing, and study skills are examples of less- structured tasks -- tasks that cannot be broken down into a fixed sequence of subtasks and steps that consistently and unfailingly lead to the goal. Because these ta sks are less-structured and difficult, they have also been called higher-level tasks. These types of tasks do not have the fixed sequence that is part of well-structured tasks. One cannot develop algorithms that students can use to complete these tasks."

Gleefully pilfered from: Barak Rosenshine, Advances in Research on Instruction, Chapter 10 in J.W. Lloyd, E.J. Kameanui, and D. Chard (Eds.) (1997) Issues in educating students with disabilities. Mahwah, N.J.: Lawrence Erlbaum: Pp. 197-221. http://epaa.asu.edu/barak/barak.html

In cognitive strategies, we include:

- heuristics (fallible but useful decision rules)
- guidelines (fallible but common descriptions of how to do things)
- good (rather than "best" practices)

The relationship between cognitive strategies and models:

- deciding to apply a model and figuring out how to apply a model involve cognitive strategies
- deciding to create a model and figuring out how to create models to represent or simplify a problem involve cognitive strategies

BUT

- the model itself is a simplified representation of something, done to give you insight into the thing you are modeling.

We aren't sure that the distinction between models and the use of them is worthwhile, but it seems natural to us so we're making it.

Variation for Testing: Models

A model is

- A simplified representation created to make something easier to understand, manipulate or predict some aspects of the modeled object or system.
- Expression of something we don't understand in terms of something we (think we) understand.
- A state-machine representation of a program is a model.
- Deciding to use a state-machine representation of a program as a vehicle for generating tests is a cognitive strategy.
- Slavishly following someone's step-by-step catalog of best practices for generating a state-machine model of a program in order to derive scripted test cases for some fool to follow is a procedure.
- This definition of a model is a concept.
- The assertion that Harry Robinson publishes papers on software testing and models is a statement of fact.

Sometimes, a rule is a model.

- A rule is an imperative ("Stop at a red light") or a causal relationship ("Two plus two yields four") or a statement of a norm ("Don't wear undershorts outside of your pants at formal meetings").
- A description / definition of the rule is probably a concept
- A symbolic or generalized description of a rule is probably a model.

Variation for Testing: Skills

Skills are things that improve with practice.

- Effective bug report writing is a skill, and includes several other skills.
- Taking a visible failure and varying your test conditions until you find a simpler set of conditions that yields the same failure is skilled work. You get better at this type of thing over time.
- Entries into this section will often be triggered by examples (in instructional materials) that demonstrate skilled work, like "Here's how I use this technique" or "Here's how I found that bug."
- The "here's how" might be classed as a:
 - procedure
 - cognitive strategy, or
 - skill
- In many cases, it would be accurate and useful to class it as both a skill and a cognitive strategy.

106

Variation for Testing: Attitudes

- "An attitude is a persisting state that modifies an individual's choices of action." Robert M. Gagne, Leslie J. Briggs & Walter W. Wager (1992) "Principles of Instructional Design" (4th Ed),, p. 48.
- Attitudes are often based on beliefs (a belief is a proposition that is held as true whether it has been verified true or not).
- Instructional materials often attempt to influence the student's attitudes.
- For example, when we teach students that complete testing is impossible, we might spin the information in different ways to influence student attitudes toward their work:
 - given the impossibility, testers must be creative and must actively consider what they can do at each moment that will yield the highest informational return for their project
 - given the impossibility, testers must conform to the carefully agreed procedures because these reflect agreements reached among the key stakeholders rather than diverting their time to the infinity of interesting alternatives
- Attitudes are extremely controversial in our field and refusal to acknowledge legitimate differences (or even the existence of differences) has been the source of a great deal of ill will.
- In general, if we identify an attitude or an attitude-related belief as something to include as an assessable item, we should expect to create questions that:
 - define the item without requiring the examinee to agree that it is true or valid
 - contrast it with a widely accepted alternative, without requiring the examinee to agree that it is better or preferable to the alternative
 - adopt it as the One True View, but with discussion notes that reference the controversy about this belief or attitude and make clear that this item will be accepted for some exams and bounced out of others.

107

Variation for Testing: Metacognition

- Metacognition refers to the executive process that is involved in such tasks as:
 - planning (such as choosing which procedure or cognitive strategy to adopt for a specific task)
 - estimating how long it will take (or at least, deciding to estimate and figuring out what skill / procedure / slave-labor to apply to obtain that information)
 - monitoring how well you are applying the procedure or strategy
 - remembering a definition or realizing that you don't remember it and rooting through Google for an adequate substitute
- Much of context-driven testing involves metacognitive questions:
 - which test technique would be most useful for exposing what information that would be of what interest to who?
 - what areas are most critical to test next, in the face of this information about risks, stakeholder priorities, available skills, available resources?
- Questions / issues that should get you thinking about metacognition are:
 - How to think about ...
 - How to learn about ...
 - How to talk about ...
- In the BBST course, the section on specification analysis includes a long metacognitive digression into active reading and strategies for getting good information value from the specification fragments you encounter, search for, or create.

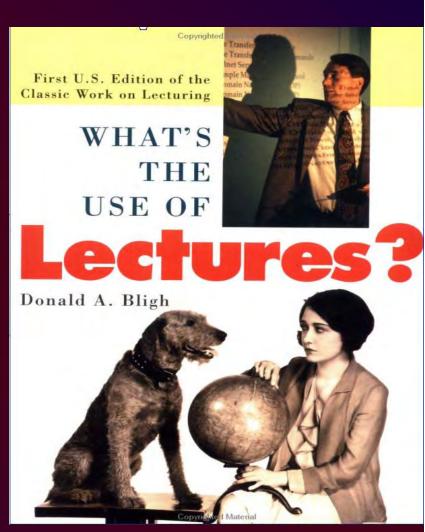
Characterizing Cognitive Complexity

		Cognitive Process Dimension					
Knowledge Dimension		Remember	Understand	Apply	Analyze	Evaluate	Create
	Factual	lecture	lecture				
	Conceptual	lecture	lecture				
	Procedural	lecture	lecture				
	Meta- Cognitive						

Anderson & Krathwohl, 2001

Commercial Teaching Style

- Primary communication style was lecture
 - Real-life examples
 - Motivating
 - Memorable
 - Illustrate applications
 - Illustrate complexity
- Lectures can be excellent for conveying basic knowledge, but they are weak for developing higher order cognitive skills



Levels of Learning

- Remember
- Understand
 - It is easy to "teach" at these levels and to assess / evaluate at them.
 - Most "objective" tests assess at these levels
- Apply
- Analyze
- Create
 - Most professional work is done at these levels.
 - We have a transfer problem. Will teaching to the lower levels transfer to the higher?

Example Problem: Domain Testing

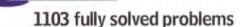
- Most widely taught testing technique
 - For details, see
 http://www.testingeducation.org/BBST/Domain.html
 - Easy to explain the basic concepts
 - Classic examples widely taught
 - Students quickly signal that they understand it
 - But when you give them exercises under slightly new circumstances
 - They blow it
 - And then they blow the next one
 - o And the next one . . .



Fourth Edition

FRANK AYRES, JR.

ELLIOTT MENDELSON



Expert tips on the graphing calculator

Many math review sections

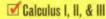
Covers all course fundamentals supplements any class text

The perfect aid for higher grades!



Brilliant (?) idea

- Lots of practice exercises
- Like we used to do as math students







Copyrighted Material

I Tried This With Commercial Students

- Many (often, most) of them needed a lot of practice under changing circumstances
- But the perceived slow pace of the course made them anxious
- And the shorter topic checklist created a marketing disadvantage for my courses.



GALGU

Fourth Edition

FRANK AYRES, JR.

ELLIOTT MENDELSON



Expert tips on the graphing calculator

Many math review sections

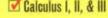
Covers all course fundamentals supplements any class text

The perfect aid for higher grades!



Back to that Brilliant (?) idea

- Lots of practice exercises
- Like we used to do as math students
- It was impractical in commercial training
- Now, at last, we can try it on university students.





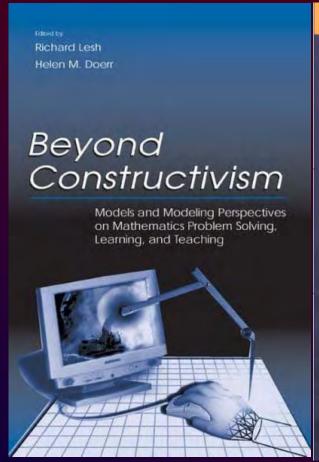


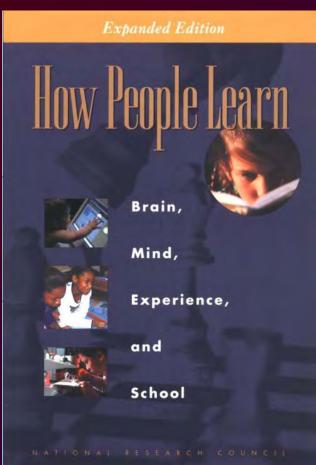
Copyrighted Material

Padmanabhan's Thesis: Practice on Domain Testing

- 15 classroom hours of lecture plus examples plus practice, practice, practice. Lots of procedural instruction and drill
- Students mastered every procedure
- Final exam
 - Applied what they knew to similar questions (near transfer)
 - They aced them
 - Applied what they knew to a problem that was beyond their practice (not beyond the lecture) (a little bit farther transfer)
 - They all failed miserably
- Successful transfer of learning requires more than procedural training and practice (This is a what-else-is-new result in science education.)

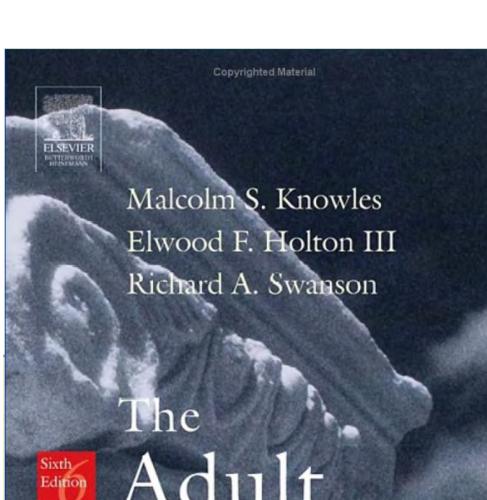
Dealing With the Transfer Problem





- In science / math education, the transfer problem is driving fundamental change in the classroom
- Students learn

 (and transfer)
 better when they
 discover
 concepts, rather
 than by being told
 them



Adult Learner

The Definitive Classic in Adult Education and Human Resource Development

Andragogy

- Pedagogy: study of teaching / learning of children
- Andragogy: study of teaching / learning of adults
- University undergrads are in a middle ground between the teacherdirected child and the fullyself-directed adult
- Both groups, but especially adults, benefit from activitybased and discovery-based styles

Open Certification

- http://www.freetestingcertification.com
- Large pool of multiple choice questions
 - We are still testing memory work, not skill, and certainly not "professionalism"
- The database of questions includes comments on each, fully open for review
- An exam is a stratified sample, downloaded free
- Administer it any time, e.g. at employment interview
- Results print the questions, the answers, and the stored discussion of the questions with "wrong" answers
- Interview discussion of the answers rather than blind reliance on a score
- Multiple flavors of exams possible, with different answers allowed as correct for same question in different exams