Adapting Academic Course Materials in Software Testing for Industrial Professional Development

More details on NSF Project:

http://www.kaner.com/pdfs/BBSTwtst2008AdvisoryBoard.pdf

More details on AST courses:

http://www.associationforsoftwaretesting.org/drupal/courses

Cem Kaner, J.D., Ph.D.
Professor of Software Engineering
Florida Institute of Technology

Abstract

Colleagues of mine in India tell me that the software testing service market there is in transition. India used to be the home of cheap software engineering labour, but wages have gone up. And as with all goods and services, there is always someone who will sell for less. Lots of service contracts that go to India now get re-outsourced elsewhere (e.g. China). There is a sense that India has about 7 years to reorganize from competing as the low-cost provider to competing as the high-skill provider. Sometimes, they draw a comparison to Japan's 1950's-1970's transition from lowest-cost manufacturing to world class. What kind of education would software testers (in the United State, India, or anywhere) need to compete with the best in their field? University education provides strong background but limited specialized instruction in software testing. And sadly, many current certification programs in software testing (like many areas of software engineering) are very popular but very superficial. The multiple-choice exam reaches to memorized knowledge more than skill. They provide an illusion of expertise, a lot of income for the organization who administers the exam and the companies who sell exam-review courses, but not so much benefit for the individual engineers or their employers. The Association for Software Testing and I are trying out a different model. We're creating a series of intensely interactive fully-online courses, each tightly focused on a specific testing skill or area of testing knowledge. Each course lasts 4 weeks. The underlying business model for this work looks a lot like the Linux model—open source (Creative Commons) course materials and a volunteer community who support the courses (AST offers them to members for free) but who also see a big service-business opportunity over the long term.

In this talk, I'll describe the first AST courses, the instructional model, and the business model, probably emphasizing how we apply a free software business model to professional education.

Students in business, software engineering, education, or psychology might find senior-project or thesis opportunities in this NSF-supported research program. We've been packaging a lot of instructional ideas together in new ways, both for academic instruction and for industrial professional development. There's a lot of data, a lot of room for trying stuff out, plenty of opportunity to publish, and a bunch of interested potential employers and consulting clients.

Context of the problem

Testing is evolving slowly because there is so little educational support for it.

- University support will continue to be inadequate for the foreseeable future.
  - Few universities offer testing courses. Fewer offer a 2nd/3rd
  - Many of the newer courses are broad and very shallow

  Companies will therefore have to develop their own training strategies.
- Commercial short courses are often ineffective because they
  - try to cover too much,
  - at too shallow a level,
  - without application to the learner's specific situation,
  - with too little opportunity for practice,
  - and less opportunity for assessment and feedback.

Commercial	Academic
Drive-by teaching: 2-5 days, rapid-fire ideas, visiting instructor	Local teaching: Several months, a few hours per week, students get to know instructor
Broad, shallow coverage	Deeper coverage
Time constraints limit activities	Activities expected to develop skills
No time for homework	Extensive homework
No exams	Assessment expected
Coached, repeated practice seen as time-wasting	Coached, repeated practice is highly appreciated
Familiarity	Capability
Work experience helps to bring home concepts	Students have no work experience, need context
Richer grounding in real practice	Harder to connect to real practice
Some (occasional) student groups share a genuine, current need	Students don’t naturally come to a course as a group with a shared problem
Objective: one applicable new idea per day	Expect mastery of several concepts and skills

What I’m up 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

The instructional challenge, as I see it

Software testing

is cognitively complex,

requires critical thinking,

effective communication, and

rapid self-directed learning.

Support for this type of learning requires intense learner engagement, which cannot be met (for most people) in passive-presentation lecture courses

Anderson Krathwohl update to Bloom's taxonomy, modified slightly for software testing

Knowledge dimension	Cognitive Process Dimension					
	Remember	Understand	Apply	Analyze	Evaluate	Create
Facts	Lecture	Lecture	Practice			
Concepts	Lecture	Lecture				
Procedures	Lecture	Lecture				
Cognitive strategies	Lecture	Lecture				
Models	Lecture	Lecture				
Skills						
Attitudes	Lecture	Lecture				
Metacognition	Lecture	Lecture				

Assessment

1. Assessment at one level (e.g. facts / concepts) is not informative with respect to another level (e.g. evaluation)
2. "Authentic assessment" – assessment with simplified or artificial tasks is uninformative with respect to what can actually be done in real circumstances (flip side of the transfer problem)
3. Assessment that is apparently at a higher level is often reducible to lower level via:
    1. Study strategies
    2. Question-answering strategies

    This is part of the strong success of exam-review courses.

The NSF Project

- The primary objective of the NSF project is adaptation and implementation of the BBST course, in several markets:
  - Academic (traditional)
  - Academic (online)
  - In-house training by in-house staff
  - Commercial training (face-to-face)
  - Commercial (online)
- To support / sustain these goals, we are working on several tasks, such as:
  - Improve the materials
  - Create supporting materials, e.g. collections of activities
  - Create a self-sustaining BBST instructor community

How the academic course works

Students watch video lecture 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

Success factors in the academic course

1. Strong content
2. Story-based teaching
3. Detailed examples
4. Video lectures
5. In-class activities that tie to the lecture
6. Application to a real product under test
7. Orientation exercises
8. Open book quizzes
9. Study-guide based exam
10. Challenging but focused assignments
11. Task scaffolding
12. Peer review
13. Explicit discussions of learning issues in the course design.
14. Open discussion of (employment) value of the material and the work
15. Organic evolution of the class (rather than process-constrained)
16. Enthusiasm and ongoing renewal (Hawthorne effect)
17. Instruction on test-taking skills
18. Student assessment of learning gains feedback

Should be success factors

I know these should make the course better, but I haven't succeeded in figuring out how:

1. Drill / problem sets, to help students
   - Experience worked examples
   - Develop skills through practice
   - Experience an underlying common core when there is a lot of more superficial variation
2. Paired testing
3. Testing competitions
4. Student presentations
5. Employer / famous-person visitors

Current challenges

1. Disappointing essay exams
2. Getting students to watch videos in advance
3. Getting students to do preparatory exercises
4. Coping with an unstandardizable vocabulary
5. Classroom time management (discussion versus lab time)
6. Videos require significant development time
7. No active discussion in the videos
8. Videos feature one white man
9. On-the-record videos make some storytelling difficult
10. Synchrony is important when students rely on each other
11. Grading time is substantial
12. Activities are hard to design
13. Multiple choice pool is small
14. We need better assigned readings
15. Student prerequisites

The Adaptation for the Association for Software Testing

Objectives

- Fully online
- Free to AST members
- Support SIGNIFICANT learning experiences

- Self-sustaining
- Commercial opportunities for instructors

Linux Model

Linux Model?

Open Source Software:

- Software is free
- Software development is
  - Free, or
  - Sponsored (e.g. IBM, Sun, HP donations)
- Services:
  - Training, installation, custom adaptation
  - Might be free, might be $$$
  - (look at Red Hat -- $$$)
- Can we do this for AST?

Open Courseware

- Course materials are free: www.testingeducation.org/BBST
- Course development is
  - Free (several volunteers, but mainly me at the moment), or
  - Sponsored (currently: NSF, Satisfice)

Open Courseware: The service model

Services?

- Teach the courses
  - instructor feedback on student work
  - answer questions, etc. VERY time consuming
- Customize the courses
  - Private versions for client companies, customized to their applications

- Can we do this for AST?
  - Free to AST members
  - AST-certified instructors
    - Intense training for certification
    - Have to agree to teach free for AST
    - Can offer course for $$$ and advertise AST cert.

So we happily set off on our first instructors course

- January 2007
- Fully online
- 18 students (North America, South America, Asia, New Zealand)
- 2 instructors (2 others who would help)
- All had significant teaching experience
  - Some university, some commercial, some K-12
- 16 had significant testing experience. Most of us had heard of each other and we were impressed with each others' reputations
- We all agreed to spend at least 8 hours / week
- And we all wanted to make this work

What could go wrong?

Instructors' Course

- Full BBST course (semester long)
- Spend a week on content
- Then spend a week on learning issues (like, how to facilitate discussions)
- Then another week on content

- Everyone was enthusiastic

We're off to see the wizard,
The wonderful wizard of…

2007 Instructors Course

... disaster

- The experts intimidated each other
  - Very politely. Big smiles.
  - People spent full day on ½ hour assignments, because the other answers were so scary.
- The quizzes drove everyone nuts
  - Especially the instructors
  - Every bad experience with teachers at school carried over
  - Every little ambiguity became an argument
- Everyone got exhausted
  - By March, we were out of steam

Lessons learned

Disasters are good

- You don’t have to wonder if a little tweak would fix it
- You don’t have to spare each other’s pride (None left to spare)
- You don’t have any tradition / precedents to worry about

So we planned for a fresh start

Fresh start

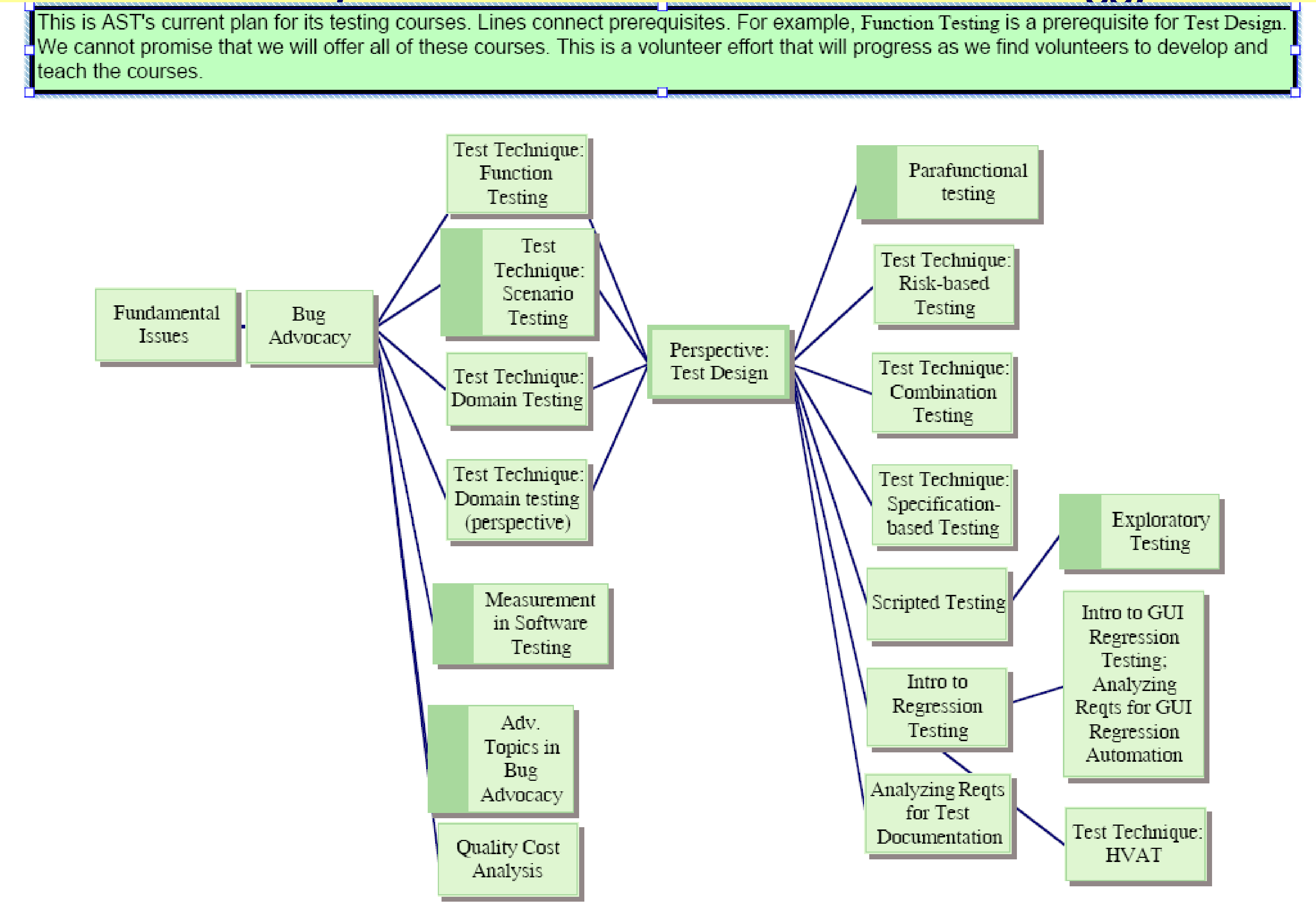
- Lots of discussions with the students on 1-on-1 basis
- Settled on a 1-month course model
  - Predictable schedules
  - Manageable commitment
- Need very careful attention to (communication about)
  - Student expectations
  - Learning objectives
  - Assessment objectives

    Online activity

    Professional disagreement

    Trust
- Settled on a narrow content model
  - One tightly integrated learning unit per course
  - Comparable to 1 week of BBST
  - Narrow content, intense application and assessment

Several narrowly defined courses, rather than 1 biggy



This is AST's current plan for its testing courses. Lines connect prerequisites. For example, Function Testing is a prerequisite for Test Design. We cannot promise that we will offer all of these courses. This is a volunteer effort that will progress as we find volunteers to develop and teach the courses.

Fundamental Issues
Bug Advocacy
Test Technique: Function Testing
Test Technique: Scenario Testing
Test Technique: Domain Testing
Test Technique: Domain testing (perspective)
Measurement in Software Testing
Adv. Topics in Bug Advocacy
Quality Cost Analysis
Perspective: Test Design
Parafunctional testing
Test Technique: Risk-based Testing
Test Technique: Combination Testing
Test Technique: Specification-based Testing
Scripted Testing
Exploratory Testing
Intro to Regression Testing
Intro to GUI Regression Testing; Analyzing Reqts for GUI Regression Automation
Analyzing Reqts for Test Documentation
Test Technique: HVAT

New course model

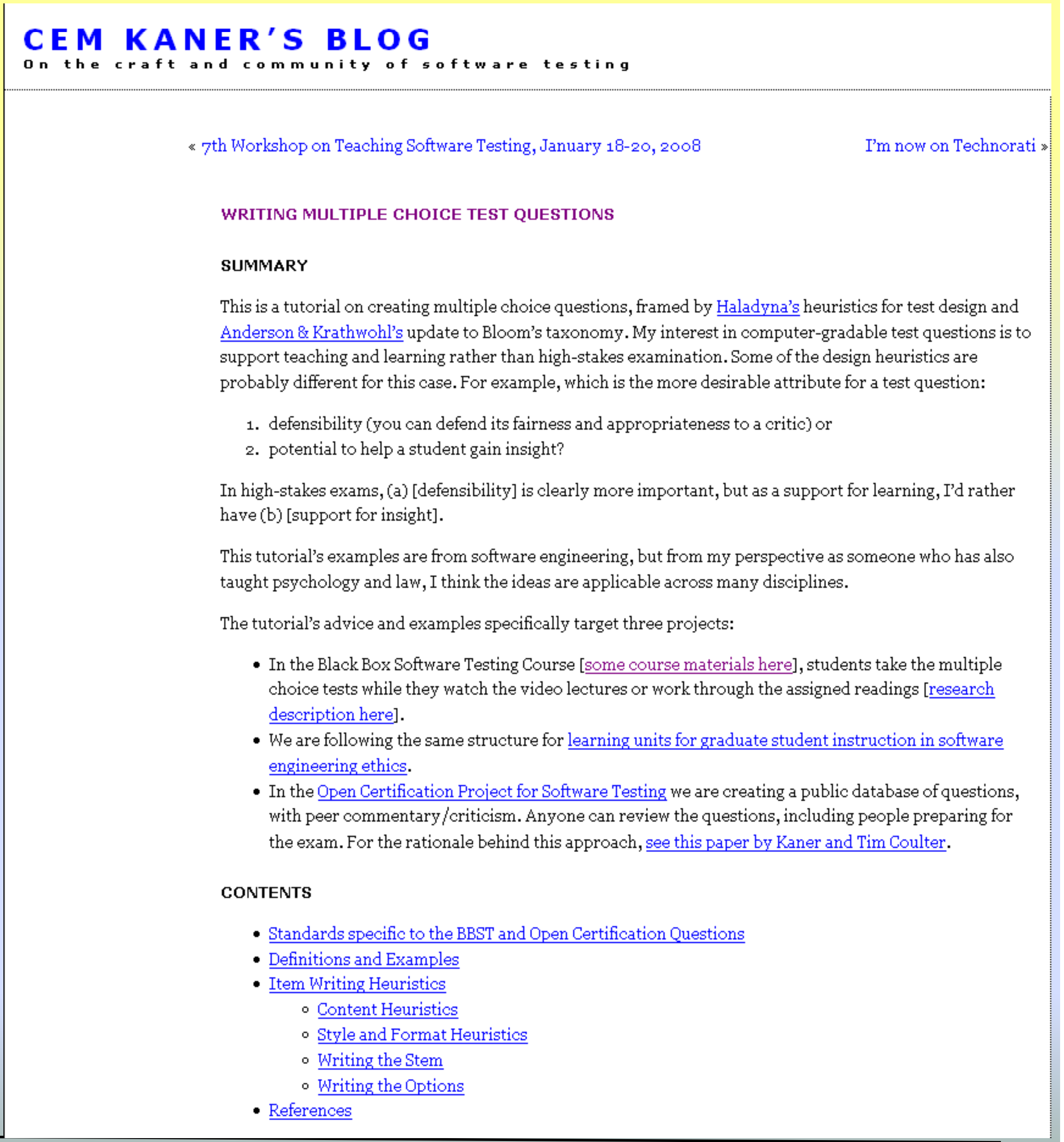
AST BBST – Foundations

Offered twice in 2007, once so far in 2008

- Informed consent (lets us do research) (IRB approval)
- 1st week
  - Get to know you discussions, pre-lecture orientation short assignments, peer review them, lectures online, quiz with each lecture (open book)
- 2nd week
  - Lectures, quizzes, begin group project(simple task, complicated by geographic diversity), begin study for exam
- 3rd week
  - Finish project, peer review project, prepare for exam
- Take-home essay exam, then peer grading

Clearer standards for multiple choice questions

- Thorough standard:
  - 15 pdf pages
- Key outcome of the open certification project, and the 2 Workshops on Open Certification



CEM KANER'S BLOG

On the craft and community of software testing

« 7th Workshop on Teaching Software Testing, January 18-20, 2008 I'm now on Technorati »

WRITING MULTIPLE CHOICE TEST QUESTIONS

SUMMARY

This is a tutorial on creating multiple choice questions, framed by Haladyna's heuristics for test design and Anderson & Krathwohl's update to Bloom's taxonomy. My interest in computer-gradable test questions is to support teaching and learning rather than high-stakes examination. Some of the design heuristics are probably different for this case. For example, which is the more desirable attribute for a test question:

1. defensibility (you can defend its fairness and appropriateness to a critic) or
2. potential to help a student gain insight?

In high-stakes exams, (a) [defensibility] is clearly more important, but as a support for learning, I'd rather have (b) [support for insight].

This tutorial's examples are from software engineering, but from my perspective as someone who has also taught psychology and law, I think the ideas are applicable across many disciplines.

The tutorial's advice and examples specifically target three projects:

- In the Black Box Software Testing Course [some course materials here], students take the multiple choice tests while they watch the video lectures or work through the assigned readings [research description here].
- We are following the same structure for learning units for graduate student instruction in software engineering ethics.
- In the Open Certification Project for Software Testing we are creating a public database of questions, with peer commentary/criticism. Anyone can review the questions, including people preparing for the exam. For the rationale behind this approach, see this paper by Kaner and Tim Coulter.

CONTENTS

- Standards specific to the BBST and Open Certification Questions
- Definitions and Examples
- Item Writing Heuristics
  - Content Heuristics
  - Style and Format Heuristics
  - Writing the Stem
  - Writing the Options
- References

Results

- ¼ dropout rate (mainly, work spikes on the job)
- Survivors report:
  - spending 12 hours / week
  - Rate the course as hard or harder than academic or commercial courses and as valuable or more than academic or commercial courses
- Course graded as complete / incomplete (pass/fail), and all instructors (3-4) had to agree that a student should fail
  - This is not university instruction
  - We want to help professionals make progress from where they are, not require them to meet a particular standard.
  - 1-2 students per class fail
- Instructors are also spending 8-12 hours per week

Course development cost

Currently doing module 2: Bug Advocacy, out for final peer review.

- Currently194 slides (lecture transcript 25962 words) spread over 6 sections
- Somewhat mature materials, several extensive revisions starting in 1987
- This revision:

– New slides	about	40 hours
– Lecture transcript	about	120 hours
– Videotaping time	predict	16 hours taping, to yield 3.5 hours video
– Video editing	predict	32 hours
– Other video production		10 hours
– Multiple guess questions		100 / 20 hours
– Update project		20 hours (10 spent)
– Orientation exercises		10 hours
– Train instructors		10 hours

- Significant rework required because of predictably high expectations of the students

Instructor development: AST Model

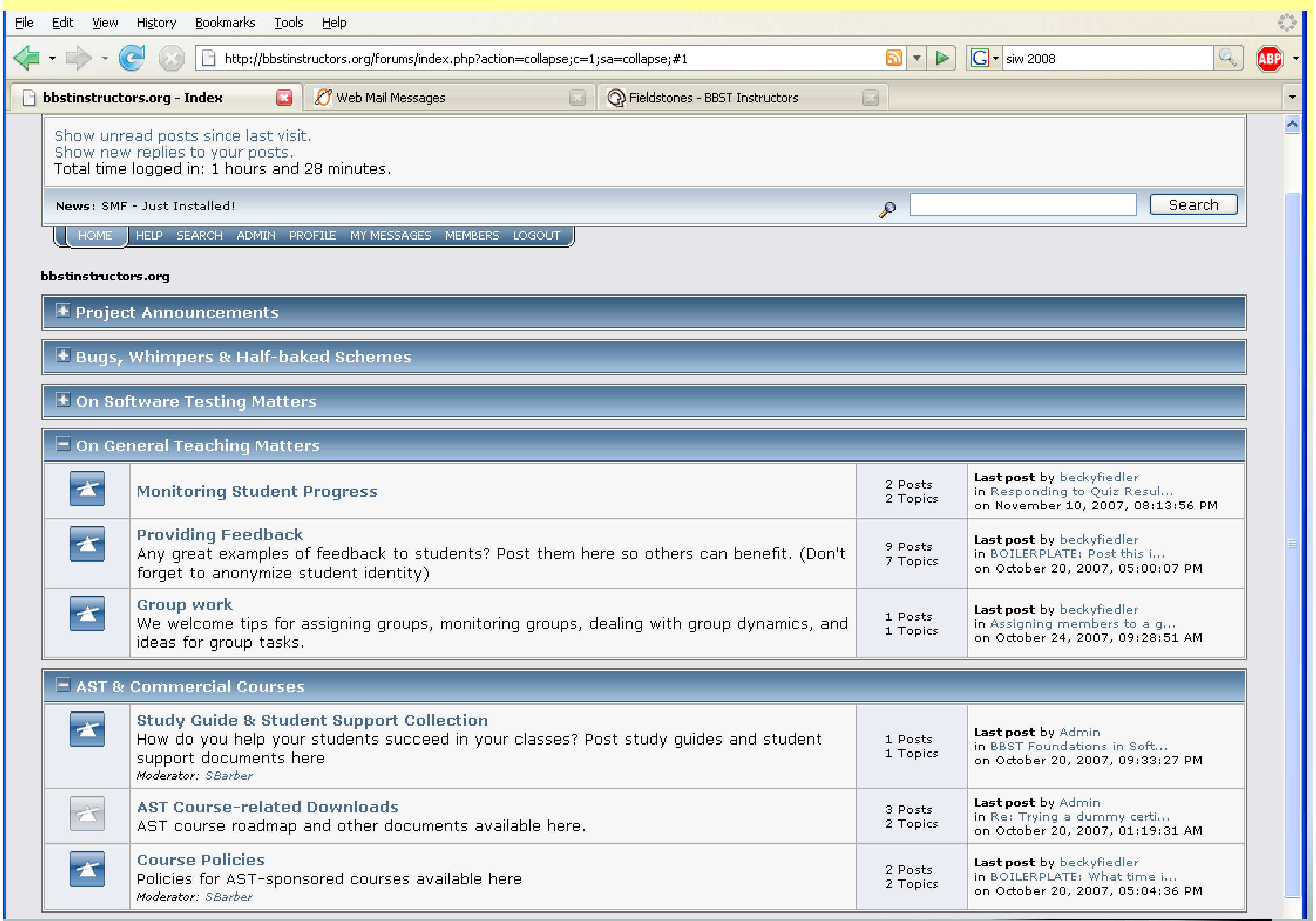
Each course has:

- Senior instructor
- Second & third instructors, who (unless the course is new) :
  - have taken the course
  - 2nd instructor was a 3rd instructor for course before
- To become senior instructor
  - Must be on third or later teaching
  - Must be supervised by a certified instructor
- To be certified for this course
  - Must successfully be senior instructor
  - Must do well on BBST instructor course
  - Must be accepted (supermajority vote) by other certified instructors for this course

Instructor support materials
www.bbstinstructors.org

We start with rough stuff at the forum…

www.bbstinstructors.org/forums


File
Edit
View
History
Bookmarks
Tools
Help
http://bbstinstructors.org/forums/index.php?action=collapse;c=1;sa=collapse;#1
siw 2008
bbstinstructors.org - Index
Web Mail Messages
Fieldstones - BBST Instructors
HOME
HELP
SEARCH
ADMIN
PROFILE
MY MESSAGES
MEMBERS
LOGOUT
Search

Show unread posts since last visit.
Show new replies to your posts.
Total time logged in: 1 hours and 28 minutes.

News: SMF - Just Installed!

bbstinstructors.org

+ Project Announcements

+ Bugs, Whimpers & Half-baked Schemes

+ On Software Testing Matters

− On General Teaching Matters

	Monitoring Student Progress	2 Posts 2 Topics	**Last post** by beckyfiedler in Responding to Quiz Resul... on November 10, 2007, 08:13:56 PM
	Providing Feedback Any great examples of feedback to students? Post them here so others can benefit. (Don't forget to anonymize student identity)	9 Posts 7 Topics	**Last post** by beckyfiedler in BOILERPLATE: Post this i... on October 20, 2007, 05:00:07 PM
	Group work We welcome tips for assigning groups, monitoring groups, dealing with group dynamics, and ideas for group tasks.	1 Posts 1 Topics	**Last post** by beckyfiedler in Assigning members to a g... on October 24, 2007, 09:28:51 AM

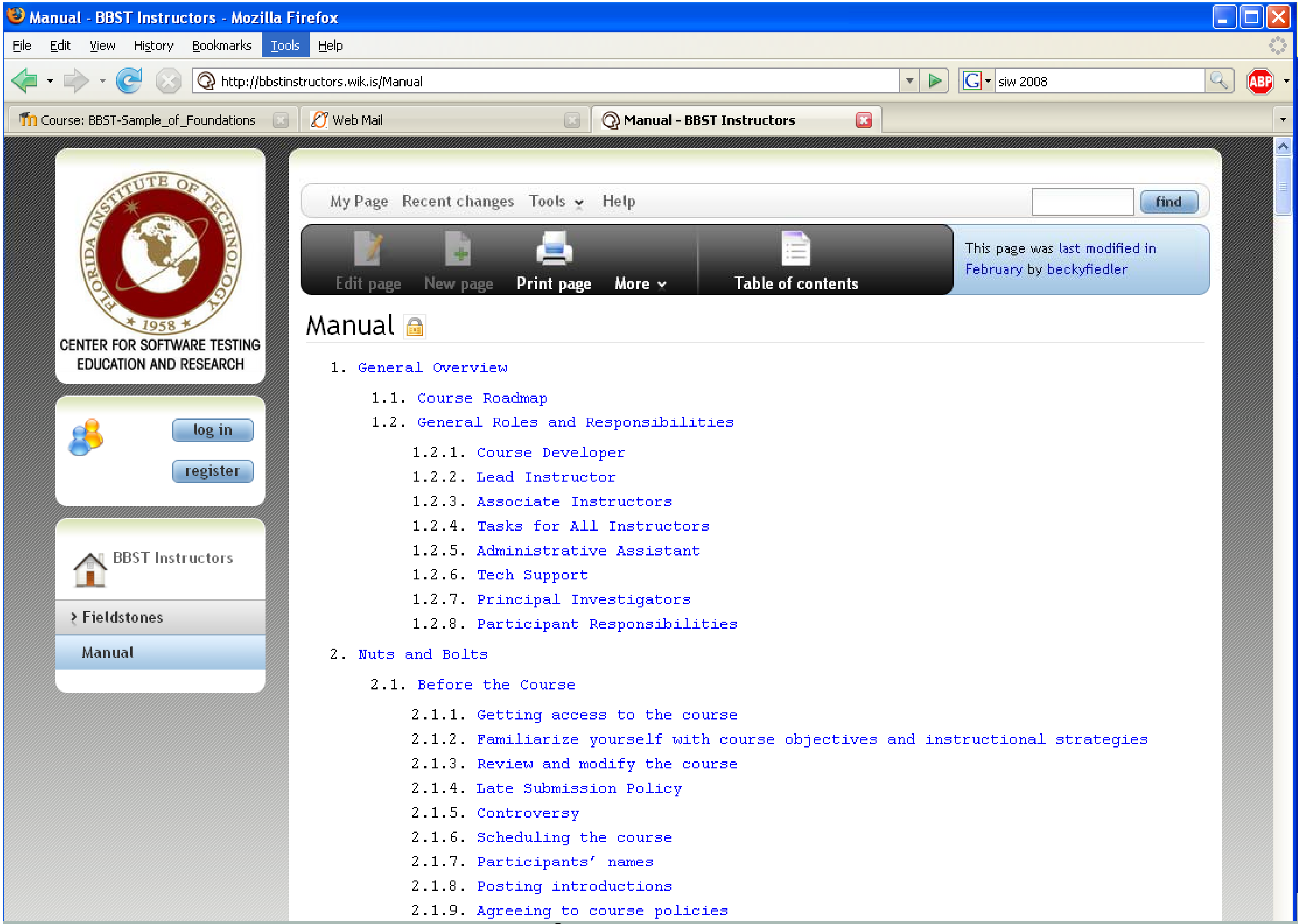
− AST & Commercial Courses

	Study Guide & Student Support Collection How do you help your students succeed in your classes? Post study guides and student support documents here *Moderator:* SBarber	1 Posts 1 Topics	**Last post** by Admin in BBST Foundations in Soft... on October 20, 2007, 09:33:27 PM
	AST Course-related Downloads AST course roadmap and other documents available here.	3 Posts 2 Topics	**Last post** by Admin in Re: Trying a dummy certi... on October 20, 2007, 01:19:31 AM
	Course Policies Policies for AST-sponsored courses available here *Moderator:* SBarber	2 Posts 2 Topics	**Last post** by beckyfiedler in BOILERPLATE: What time i... on October 20, 2007, 05:04:36 PM

Polished stuff goes to the wiki

bbstinstructors.wiki.is/Manual

Instructor manual


Manual - BBST Instructors - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://bbstinstructors.wik.is/Manual
siw 2008
Course: BBST-Sample_of_Foundations
Web Mail
Manual - BBST Instructors
INSTITUTE OF TECHNOLOGY FLORIDA 1958
CENTER FOR SOFTWARE TESTING EDUCATION AND RESEARCH
log in
register
BBST Instructors
Fieldstones
Manual
My Page Recent changes Tools Help
find
Edit page
New page
Print page
More
Table of contents

This page was last modified in February by beckyfiedler

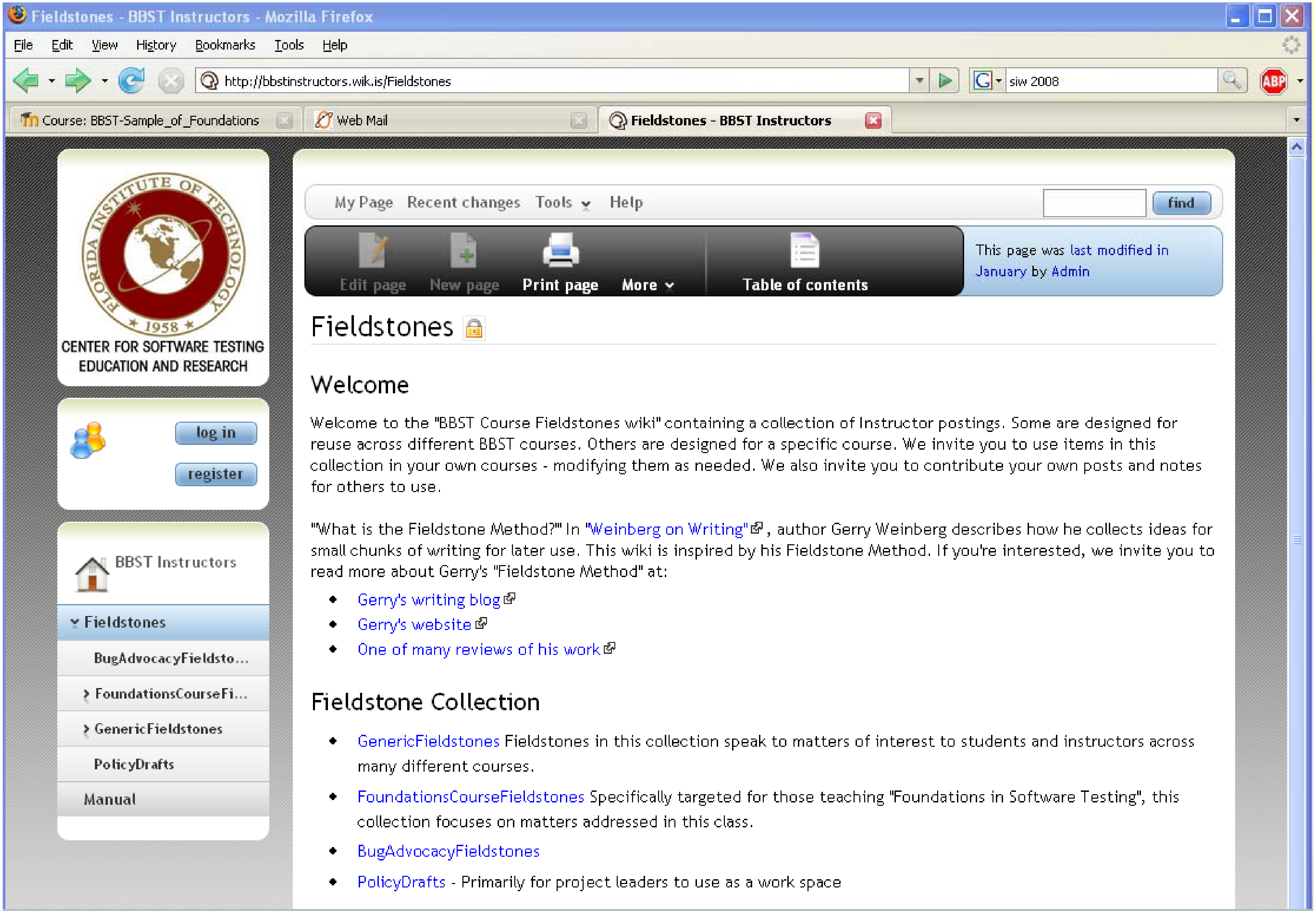
Manual

1. General Overview
   1.1. Course Roadmap
   1.2. General Roles and Responsibilities
      1.2.1. Course Developer
      1.2.2. Lead Instructor
      1.2.3. Associate Instructors
      1.2.4. Tasks for All Instructors
      1.2.5. Administrative Assistant
      1.2.6. Tech Support
      1.2.7. Principal Investigators
      1.2.8. Participant Responsibilities
2. Nuts and Bolts
   2.1. Before the Course
      2.1.1. Getting access to the course
      2.1.2. Familiarize yourself with course objectives and instructional strategies
      2.1.3. Review and modify the course
      2.1.4. Late Submission Policy
      2.1.5. Controversy
      2.1.6. Scheduling the course
      2.1.7. Participants' names
      2.1.8. Posting introductions
      2.1.9. Agreeing to course policies

Polished stuff goes to the wiki

bbstinstructors.wiki.is/Fieldstones

Fieldstones are text fragments worthy of reuse from course to course


Fieldstones - BBST Instructors - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://bbstinstructors.wik.is/Fieldstones
siw 2008
Course: BBST-Sample_of_Foundations
Web Mail
Fieldstones - BBST Instructors
CENTER FOR SOFTWARE TESTING EDUCATION AND RESEARCH
log in
register
BBST Instructors
Fieldstones
BugAdvocacyFieldsto...
FoundationsCourseFi...
GenericFieldstones
PolicyDrafts
Manual
My Page
Recent changes
Tools
Help
find
Edit page
New page
Print page
More
Table of contents

This page was last modified in January by Admin

Fieldstones

Welcome

Welcome to the "BBST Course Fieldstones wiki" containing a collection of Instructor postings. Some are designed for reuse across different BBST courses. Others are designed for a specific course. We invite you to use items in this collection in your own courses - modifying them as needed. We also invite you to contribute your own posts and notes for others to use.

"What is the Fieldstone Method?'" In "Weinberg on Writing", author Gerry Weinberg describes how he collects ideas for small chunks of writing for later use. This wiki is inspired by his Fieldstone Method. If you're interested, we invite you to read more about Gerry's "Fieldstone Method" at:

- Gerry's writing blog
- Gerry's website
- One of many reviews of his work

Fieldstone Collection

- GenericFieldstones Fieldstones in this collection speak to matters of interest to students and instructors across many different courses.
- FoundationsCourseFieldstones Specifically targeted for those teaching "Foundations in Software Testing", this collection focuses on matters addressed in this class.
- BugAdvocacyFieldstones
- PolicyDrafts - Primarily for project leaders to use as a work space

We’ll evolve this into a moodle-based instructor course

- self paced
- not facilitated
- save your own assignments for review if you apply to be an AST certified instructor

Where we're going: AST

1. New certification model
   - Take 10 AST courses, get a certificate
   - (similar to UC Extension, except this is narrowly focused)
   - About 500 hours work, probably spread over 2-3 years
   - Contrasts with ISTQB, QAI, ASQ multiple-guess certification exams that you can cram for with a 1-3 day review course
2. Live 1-day instructor course at CAST (conference of the AST) this summer, help us certify more instructors
   - Currently, 4 candidates in training, expect 6 or 7 by July
   - Probably 30-50% of instructors will get certified
3. First commercial offering of AST Foundations, probably this fall (after first formal certification)

Where we're going: NSF Project

- Opportunities for:
  - Broad collaboration across industry / academic and across academic institutional boundaries
  - Broad data pool
  - Grants to support collaboration and assessment
  - Commercial profit
  - A broader pool of people achieving technician-level or higher-level entry into the skilled workforce
- More general benefits
  - Activities pool more broadly useful in SE education
  - Instructional methods more broadly applicable

Project assessment

- Project task tracking
- SALG characterizations
- **Blind comparisons of final exams across courses (how do the answers rank, across courses)**
- **Open comparisons of final exams across courses (how are the answers similar or different across courses)**
- Employer reactions X months later
- Student reactions X months later
- Instructor reactions
- Adoption statistics?
- Peer review / external evaluation
- ??? What else ???

What the project needs to work on

- Additional venues (including access to assessment data)
  - Huston-Tillotson, U Illinois Springfield, Dalhousie, community college, Quardev (independent lab), several others
- Figuring out what assessment data we should collect
- Actually doing the analyses
- New models for video content (and doing the videos)
  - Discussion videos to supplement lecture
  - New lecturers
- Publishable pools of activities
- The Oxford English analogy for documenting testing vocabulary
- Funding to support additional venues
- Funding to support administration of the project
- Active collaborators on the instructors’ course
- Writing up what we’re learning

About Cem Kaner

- Professor of Software Engineering, Florida Tech
- Research Fellow at Satisfice, Inc.

I've worked in all areas of product development (programmer, tester, writer, teacher, user interface designer, software salesperson, organization development consultant, as a manager of user documentation, software testing, and software development, and as an attorney focusing on the law of software quality.)

Senior author of three books:

- *Lessons Learned in Software Testing* (with James Bach & Bret Pettichord)
- *Bad Software* (with David Pels)
- *Testing Computer Software* (with Jack Falk & Hung Quoc Nguyen).

My doctoral research on psychophysics (perceptual measurement) nurtured my interests in human factors (usable computer systems) and measurement theory.