Building a Free Courseware Community Around an Online Software Testing Curriculum

Cem Kaner kaner@kaner.com
Professor of Software Engineering
Florida Institute of Technology

Rebecca L. Fiedler fiedler@beckyfiedler.com
Assistant Professor of Education
Indiana State University

Scott Barber sbarber@perftestplus.com
Executive Director
Association for Software Testing

http://conference.merlot.org/2008/Saturday/kaner_c_Saturday.pdf

The Fundamental Goal

Improve the state of the practice in software testing

By improving the quality and availability of software testing education

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

Traditional models aren't working

- 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.

Academic Course	Commercial Short Course
Local instructor, who the student interacts with several times. Students get to know the instructor.	Visiting instructor (or a stranger in a conference / training room in a hotel). Few students get to know the instructor.
Spread over several months. Students have time to question and digest the material.	Rapid-fire ideas over a few days.
Deeper coverage of the materials	Broader, shallower coverage
Many courses emphasize activities expected to develop skills	Time constraints limit activities. Most courses rely heavily on lecture
Extensive homework	No time for homework
Students expect assessment (e.g. assignments and exams that are not trivially easy)	No exams, or a relatively easy multiple-choice exam
Coached, repeated practice is highly appreciated, especially if this material might appear on an exam	Coached, repeated practice seen as time-wasting. Coverage (more material) is more important than mastery. The expectation is that if a student sees that an idea or area is interesting, s/he will investigate it later, on her own time.

Academic Course	Commercial Short Course
The goal is to develop capability (can the student DO this?)	The goal is to develop familiarity: Does this student know about this elect
Students have no work experience, need context	Work experience helps to bring home concepts
Harder to connect to the course to real practices in the field in a way that hits home for the students	Students have grounding in real practice and compare the course lessons with their experiences.
Students don't naturally come to a course as a group with a shared problem and therefore there is no natural application or task that all of them will want to solve.	Some (occasional) student groups share a genuine, current need. If all the students are in the group, the instructor can customize the course to help them with their issues.
Expect mastery of several concepts and skills	Objective: a few useful ideas that the student will consider applying on the job and exposure to a broadening set of definitions and ideas.

A new approach: Online professional development

Free self-study

- www.testingeducation.org/BBST
- video lectures, slides, etc. available to everyone
- BUT
  - no personalized guidance
  - no coached activities that apply the material
  - no assessment of student's knowledge, etc.

Instructor-led

- Association for Software Testing
- 4-week courses
- Instructor-led
- Lots of instructional support:
  - Coached activities
  - Tests / exams / discussion groups

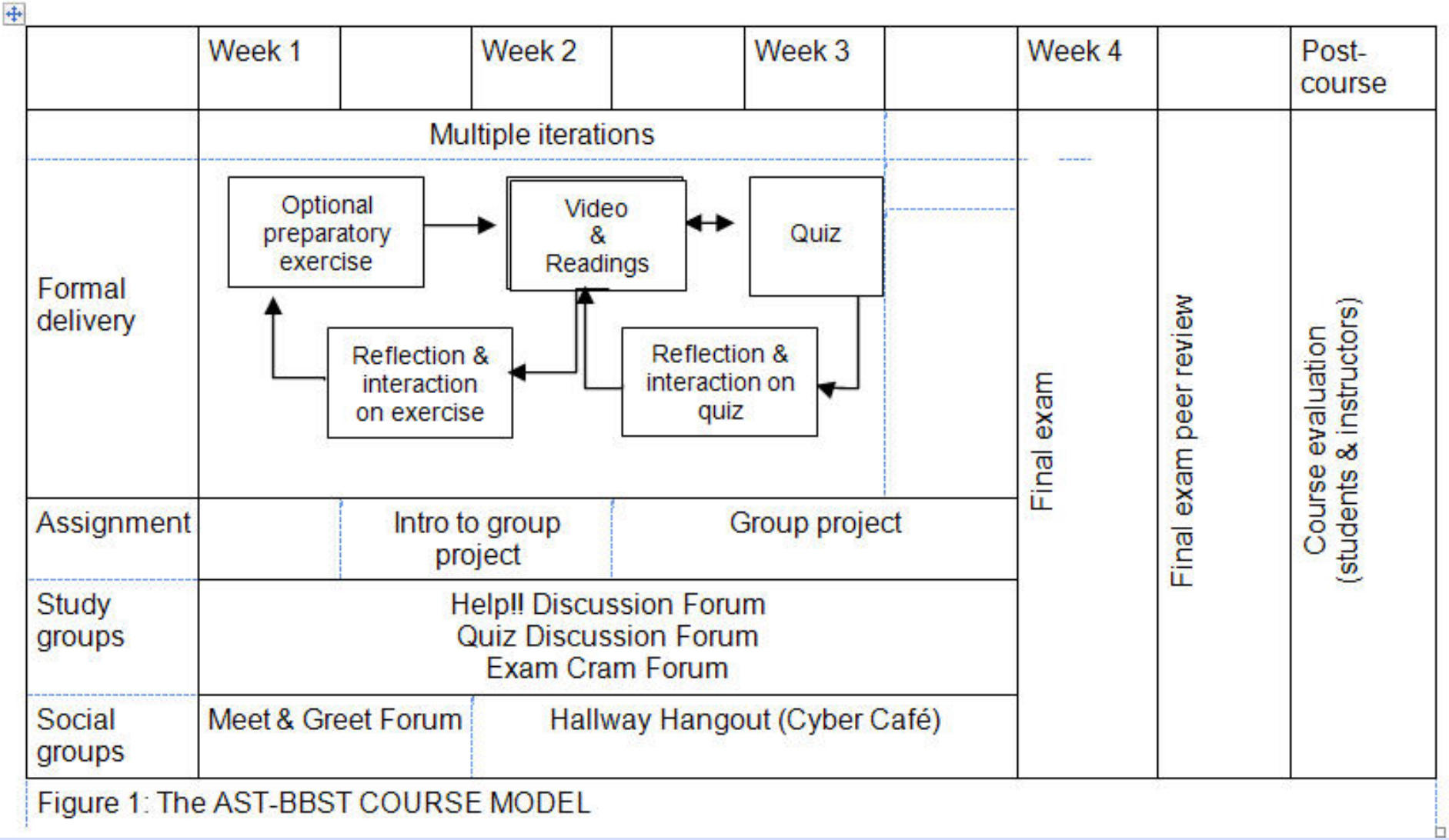


Figure 1: The AST-BBST COURSE MODEL

But, but

Developing these courses is expensive.

- 300 hours or more

Running the courses is also expensive.

- Typical student spends 12 hours / week for 4 weeks
- Typical AST course has 3 instructors
- Typical instructor spends 8-12 hours per week for 6 weeks (week before, week after)

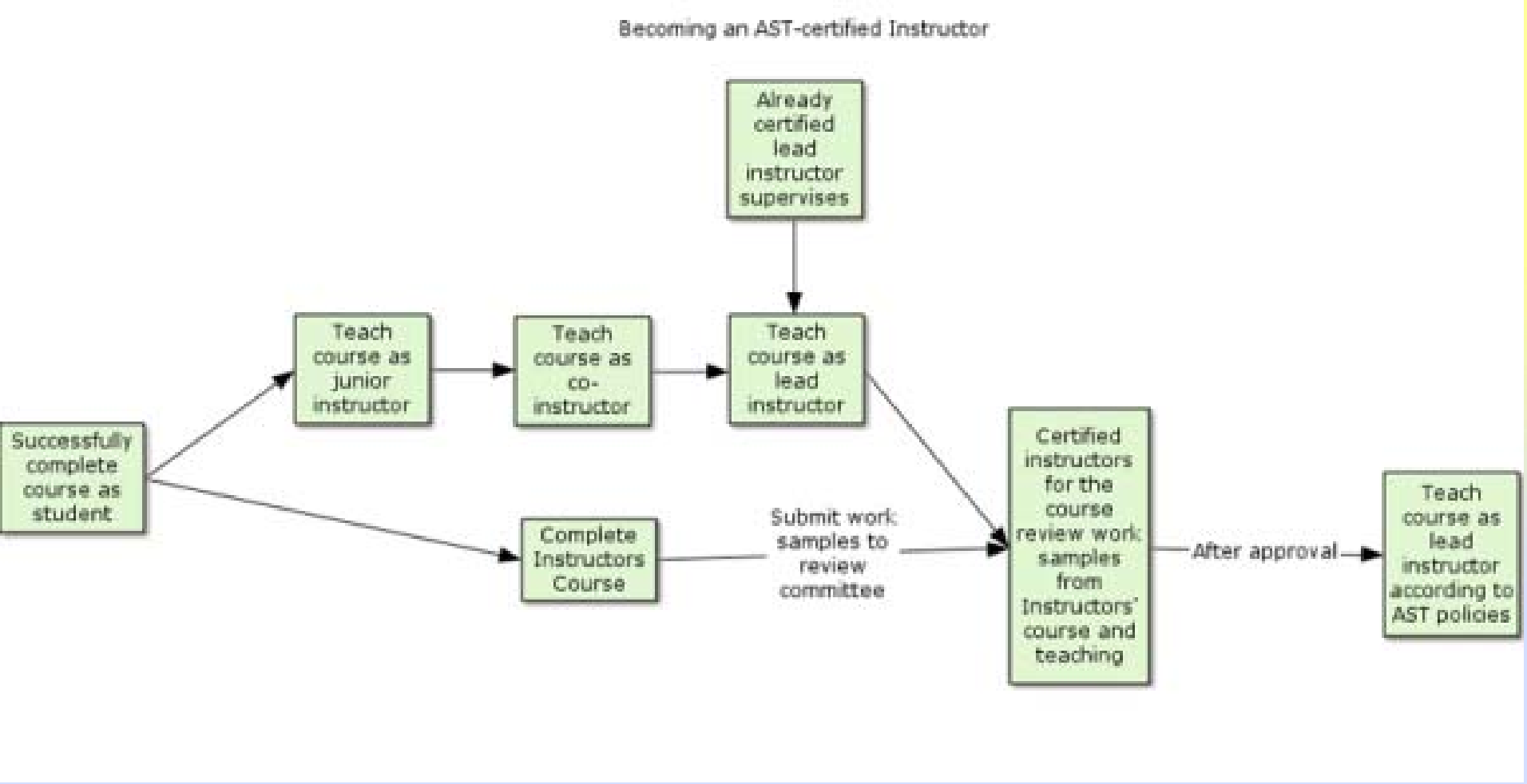
- How do we afford to create the courses?
  - What about maintenance?
- How do we attract / train volunteers to teach the courses?
  - How long will they do this?

Open Source Software Development

- Similar problems:
  - Easy to put something on SourceForge, but
    - who joins the project and why?
    - how much training of them before they deliver more than they cost?
- Consulting model
  - donate labor into the community
  - build credibility / expertise
  - clients get the software for free, but pay for services such as installation / training / customization / extension
- Can we apply this to "open source" courseware development?

Thinking through the open courseware model

- **Training model.** How should we train people?
- **Adoption model.** Why should people take these courses or use the courseware?
- **Development model.** Who will develop / approve the courses?
- **Recruitment model.** How should we recruit / qualify potential instructors?
- **Instructor training model.** How should we train / evaluate / certify potential instructors?
- **Business model.** Who should get paid, when and for what?
- **Funding infrastructure.** How can we pay for equipment, platform, development, etc.?
- **Maintenance model.** Who updates slides, videos, questions, etc.?
- **Instructor retention.** How can we retain trained instructors and keep them teaching?
- **Intellectual property.** Who should own the courseware? Can anyone maintain it?


Becoming an AST-certified Instructor
Already certified lead instructor supervises
Teach course as junior instructor
Teach course as co-instructor
Teach course as lead instructor
Successfully complete course as student
Complete Instructors Course
Submit work samples to review committee
Certified instructors for the course review work samples from Instructors' course and teaching
After approval
Teach course as lead instructor according to AST policies

Thinking through the open courseware model

- **Training model.** How should we train people?
- **Adoption model.** Why should people take these courses or use the courseware?
- **Development model.** Who will develop / approve the courses?
- **Recruitment model.** How should we recruit / qualify potential instructors?
- **Instructor training model.** How should we train / evaluate / certify potential instructors?
- **Business model.** Who should get paid, when and for what?
- **Funding infrastructure.** How can we pay for equipment, platform, development, etc.?
- **Maintenance model.** Who updates slides, videos, questions, etc.?
- **Instructor retention.** How can we retain trained instructors and keep them teaching?
- **Intellectual property.** Who should own the courseware? Can anyone maintain it?

Acknowledgements

This work was partially supported by NSF Grants EIA-0113539 ITR/SY+PE "Improving the education of software testers" and 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 authors and do not necessarily reflect the views of the National Science Foundation.

The authors would also like to acknowledge the collaboration of James Bach, Hung Quoc Nguyen, and Doug Hoffman in the development of the underlying course materials and the encouragement and support of Dr. William Shoaff, Department Chair at Florida Tech and Michael Kelly, former President of the Association for Software Testing.

More details on NSF Project:

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

More details on AST courses:

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

