UNIVERSITY of WASHINGTON

Provisioning Secure Coding Curricular Resources: Toward Robust Software

Barbara Endicott-Popovsky *and Sam Chung*

Center for Info. Assurance & Cybersecurity

University of Washington

Viatcheslav Popovsky

University of Idaho

Courtesy: NSF Award 0912109

The 16th Colloquium for Information Systems Security Education Orlando FL June 11-13, 2012

Agenda

- Why Secure Coding Practices?
- Secure Coding Teaching Models
- Motivation
- Problem Statement
- Secure Code Workshop
- Workshop Curriculum Components
- Key Findings from the Workshop
- Assessment
- Lessons Learned and Future Work

Why Secure Coding Practices?

- 90 % of reported security incidents result from exploits against defects in the design or code of commonly used software
  (Endicott-Popovsky, B.E., Frincke, D., V.M. Popovsky. At CISSE 2005)

- According to Symantec's vulnerability trend analysis, the total number of vulnerabilities is on the rise, from 4,814 in 2009 to 6,253 in 2010—a 30% increase.
  (Symantec's Vulnerability Trends, Accessed on 3/2/2012)

Secure Coding Teaching Models

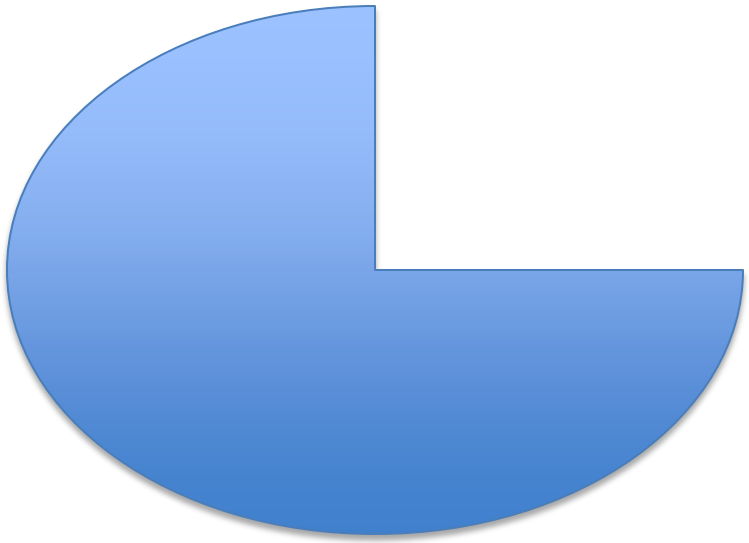
- The single course approach
- The track approach
- **The thread approach**

Bolted on

Baked in

Motivation

- Others are unsure about how to incorporate secure coding concepts into existing courses.
- Still others are simply unaware.



Problem Statement

- What are good solutions for attempting the thread approach?

Secure Code Workshop

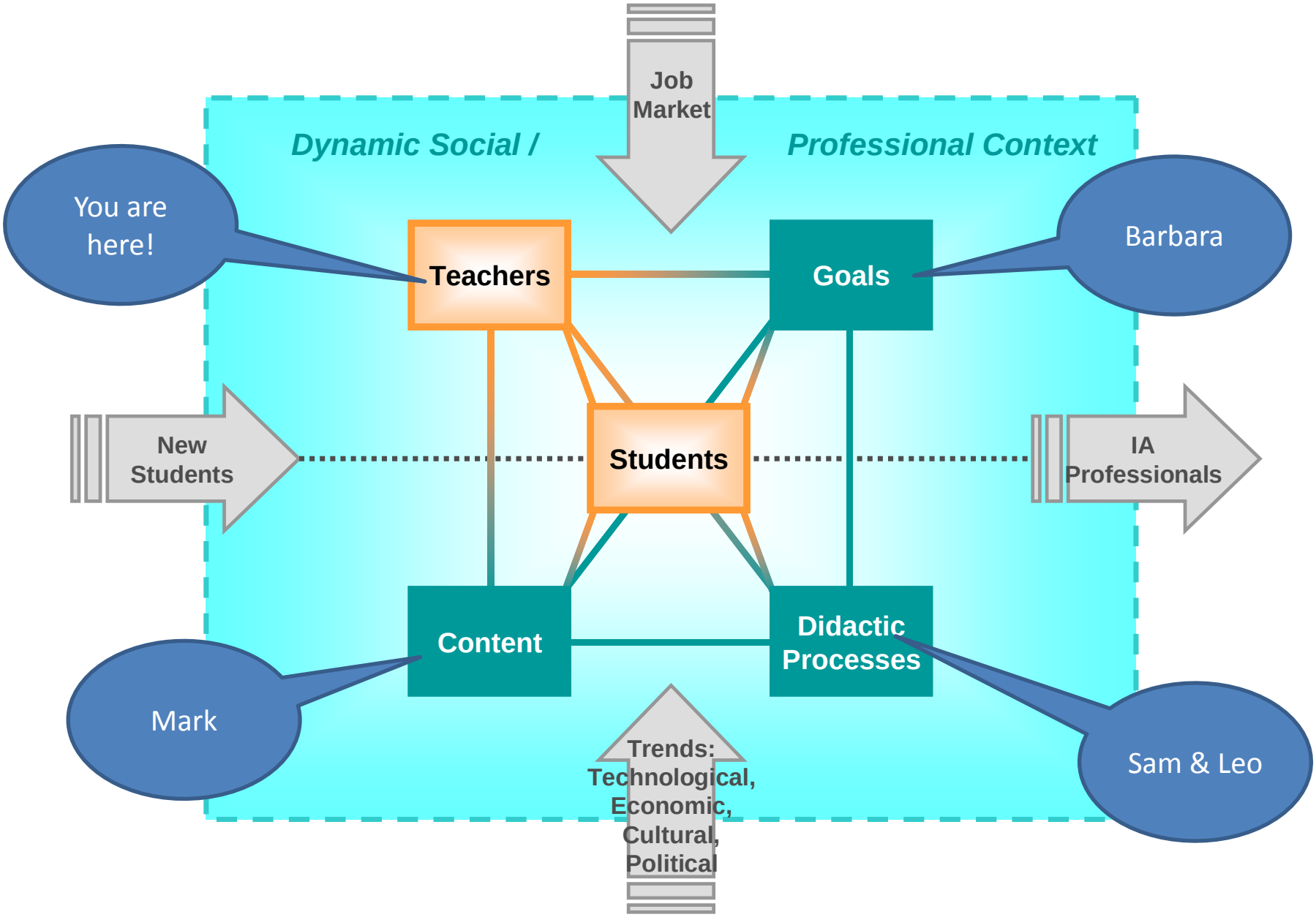
- Two days
- Nine instructors from several schools
- Each taught programming classes at either the beginning or intermediate levels.
- Workshop components - Abstract concepts to hands-on exercises.
- Internal and external evaluations
- Each workshop component was individually assessed.
- Open-ended suggestions

Workshop Curriculum Components

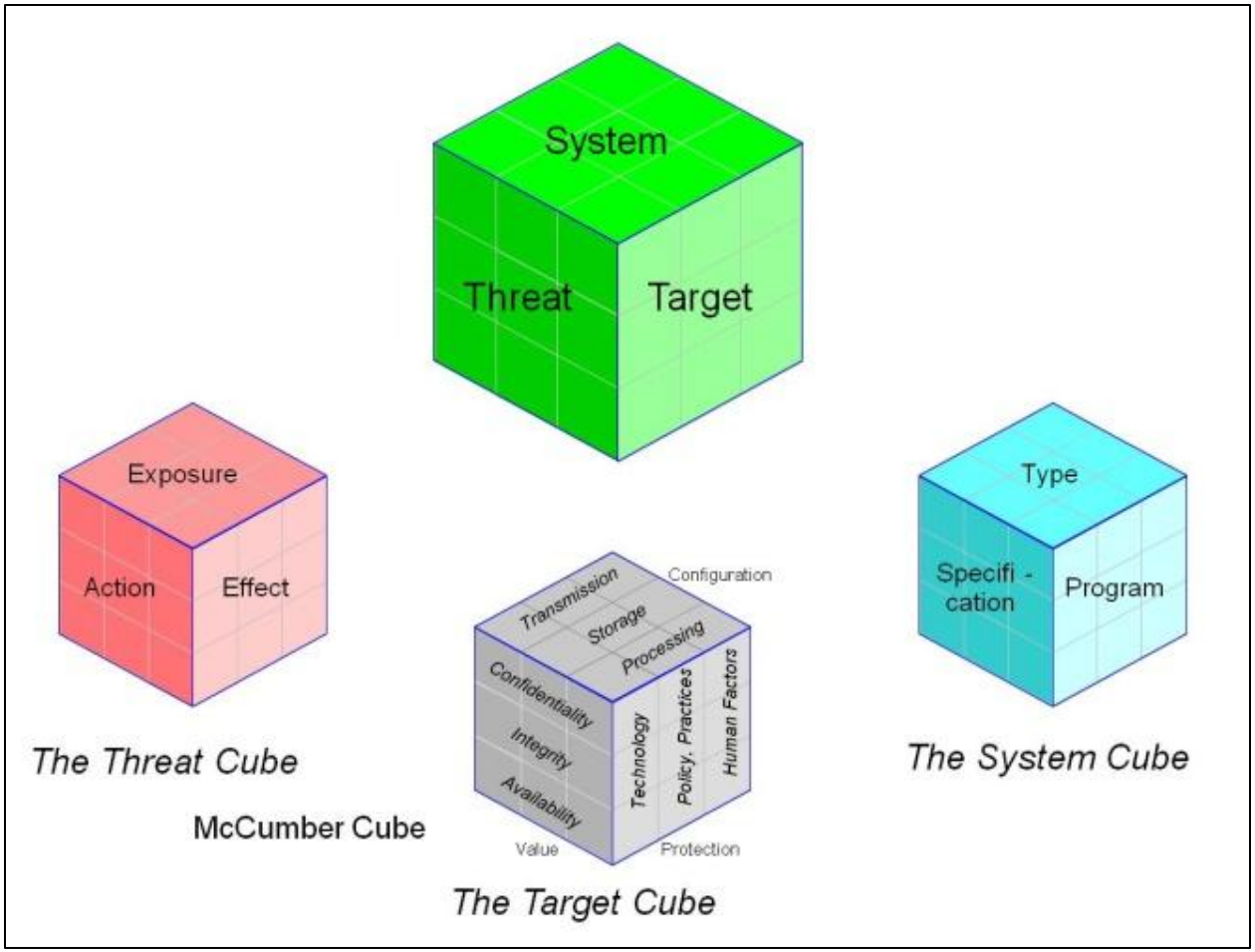
- Pedagogical Model
- Asset Projection Model (APM)
- Secure Development Lifecycle (SDL)
- Threat Modeling Tool (TMT)
- Software Reengineering Based Secure Coding
- Example Cases

KBP

Pedagogical Model for IA Curriculum Development



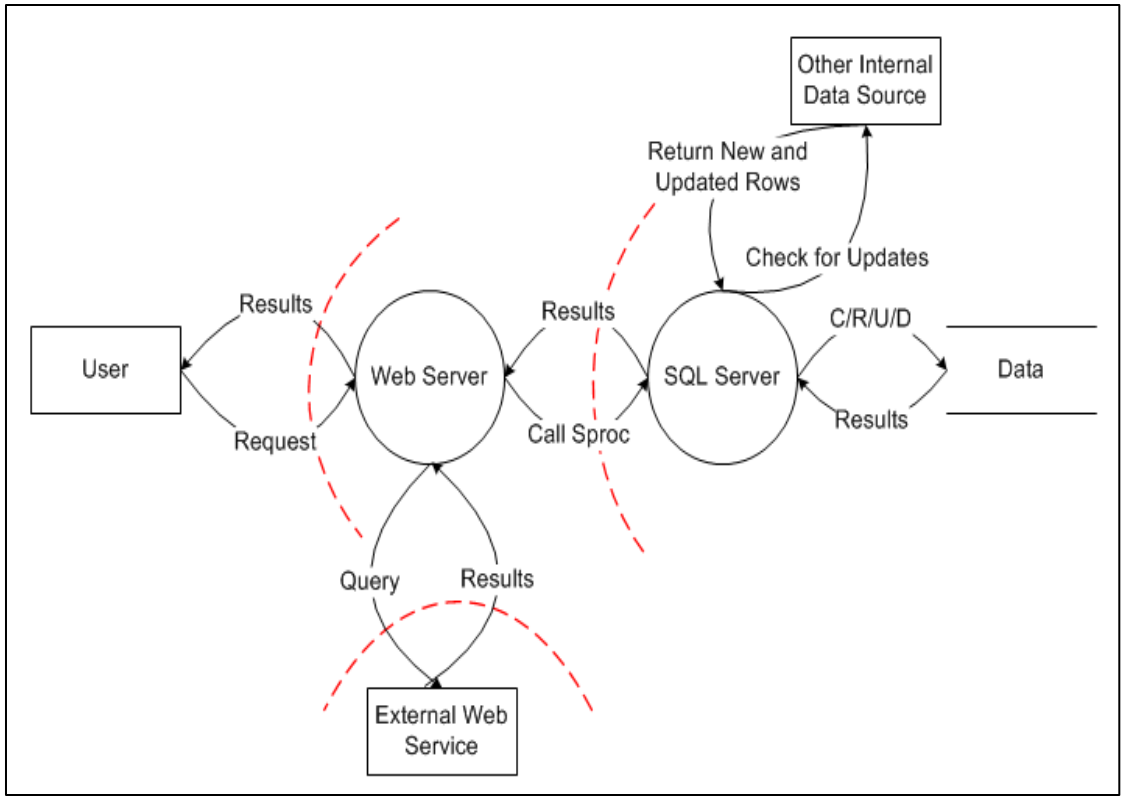
Asset Projection Model (APM)



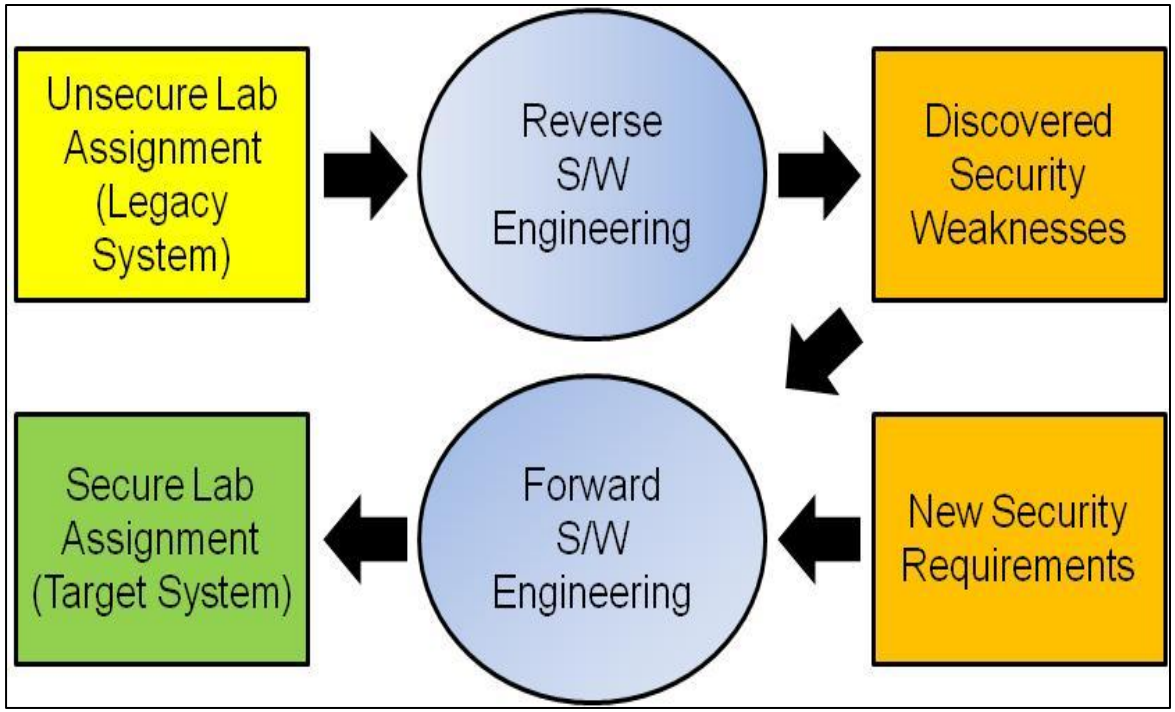
Secure Development Lifecycle (SDL)

- Systems Security Engineering Capability Maturity Model (SSE-CMM) by the Software Engineering Institute at Carnegie Mellon University
- Software Assurance Maturity Model (SAMM) developed by Open Web Applications security Project (OWASP)
- **Microsoft's Security Development Lifecycle (SDL)**

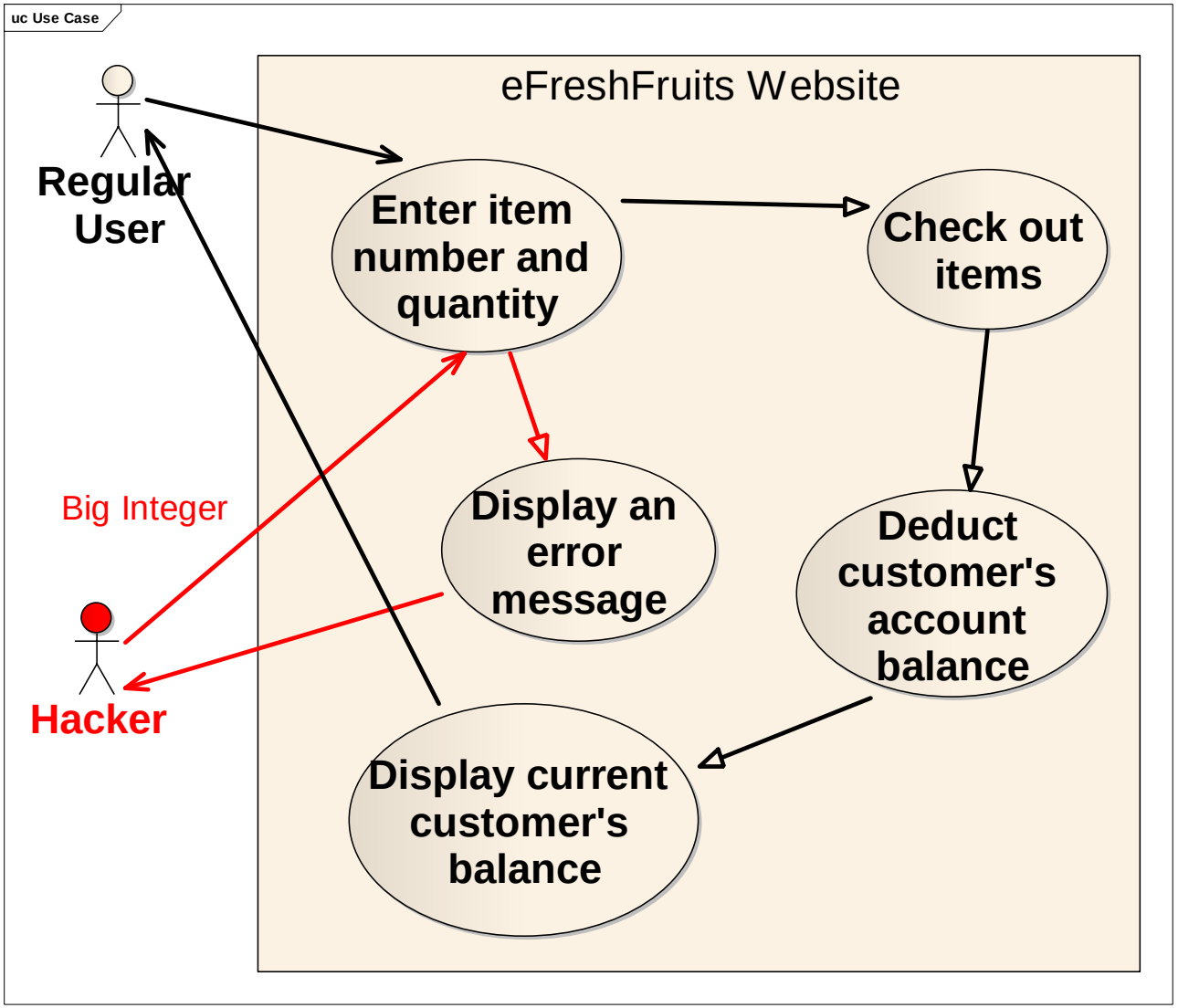
Threat Modeling Tool (TMT)



Software Reengineering Based Secure Coding

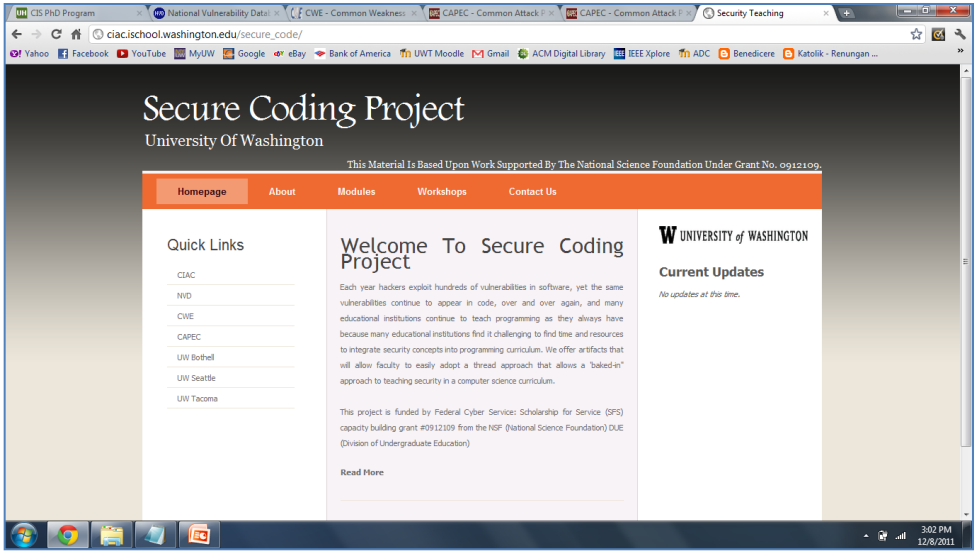


UML Diagrams



Example Cases

1. **Cross Site Scripting**
2. **SQL Injection**
3. **Buffer Overflow**
4. Integer Overflow
5. Authentication and Authorization
6. XML Injection
7. Information Exposure
8. API Abuse



http://ciac.ischool.washington.edu/secure_code/

Key Findings from the Workshop

- Background of Participants
- CS /IS Course Curricula Represented
- Goals for Learning Secure Coding Tools

Background of Participants

- Nine instructor participants
- Either self-selected based on faculty announcements, or personally solicited by the organizers.
- As a result, participants had more than a passing interest in secure code.
- The workshop was a pilot for a series of workshops to be offered the following summer.

Background of Participants

- Attendees had worked in both industry and academia.
- Industry experience ranged from 3 to 30 years—averaging 14 years each.
- Instructor experience ranged from two quarters to 25 years—also averaging 14 years each.
- Six described their primary students as upper division, graduate, or professional returning adults.
- Two taught primarily first or second year undergraduate students.
- Both 4-year (7 faculty—5 from research schools, 2 from teaching-oriented schools) and 2-year institutions (2 faculty) were represented.

CS /IS Course Curricula Represented

- Participants reported teaching a wide variety of computer and information science topics.
- Some also taught general classes.
- Two taught entire courses in security, one actually taught a course in secure development and wanted to learn additional techniques.
- Six incorporated secure coding concepts in their courses.

Goals for Learning Secure Coding Tools

- Five observed that, as novices, students have difficulty grasping the implications of code vulnerabilities or inherent vulnerabilities in client-server architecture and networking.
- Those who already taught some security concepts had been motivated by observing security challenges that continually occurred in student assignments.
- Another identified being motivated by a desire to reduce vulnerabilities in deployed code.
- Another wanted to influence students to adopt personal computer security measures.

Goals for Learning Secure Coding Tools

- At the end of the workshop, seven indicated intent to incorporate the workshop materials in some fashion.
- Those inspired to teach security in the future were motivated by a desire to provide students with a set of secure code best practices.
- Participants noted that they would add security topics to their current teaching.
- While seven indicated their curriculum would change as a result of the workshop, one said that theirs would not and the other gave a more noncommittal response.

Assessment

- Workshop Assessment
- Barriers to Implementation
- Workshop Component

Workshop Assessment

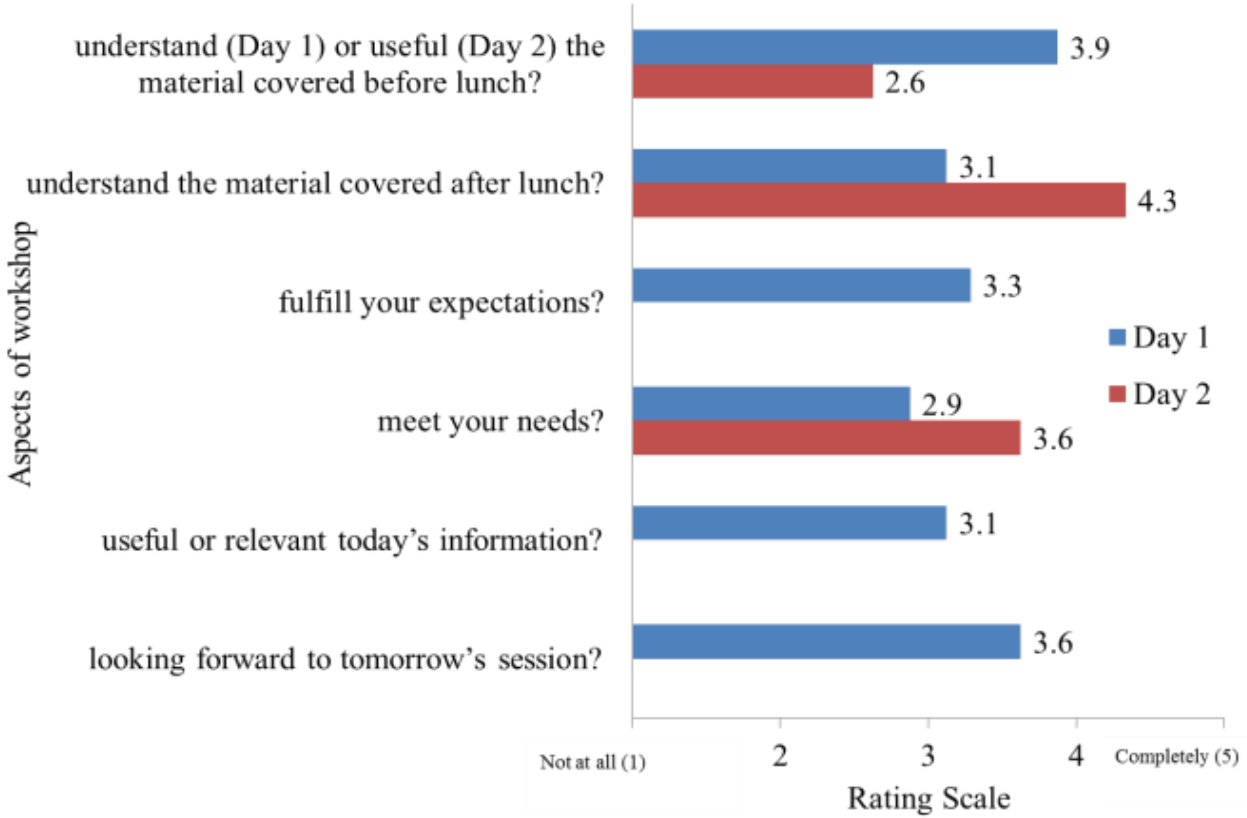


Figure 5: Workshop Assessment

Workshop Assessment

- What could be done better, they suggested:
  - More examples of code (especially Java)
  - More classroom materials
  - Packaged validation functions for JavaScript/ PHP
  - Packaged examples of XSS & SQL injection in JavaScript/PHP
  - Resources made available on the web
  - Additional, ongoing training

Barriers to Implementation

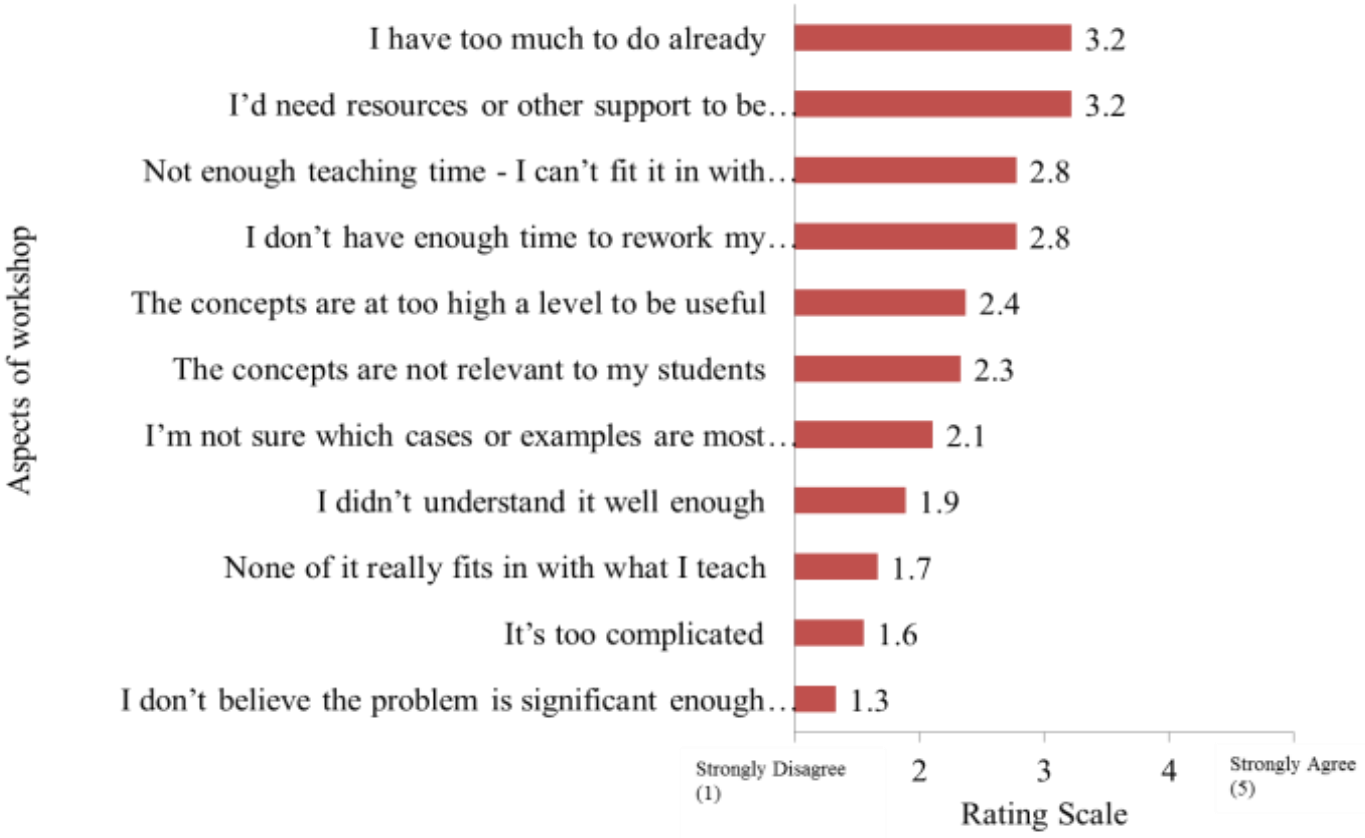


Figure 6: Potential Barriers to Implemenation

Barriers to Implementation

- The three next most challenging barriers were
  - **The need for more resources,**
  - **The lack of teaching time within the curriculum**
  - **The lack of time to make the needed changes**.
- Two participants indicated that these challenges were great enough to prevent implementation.
- One thought that the concepts were at too high a level to be useful along with not having enough teaching time to fit the material in with other required topics.

Workshop Component

How Well Understood?
How Relevant?
How Likely to Implement?

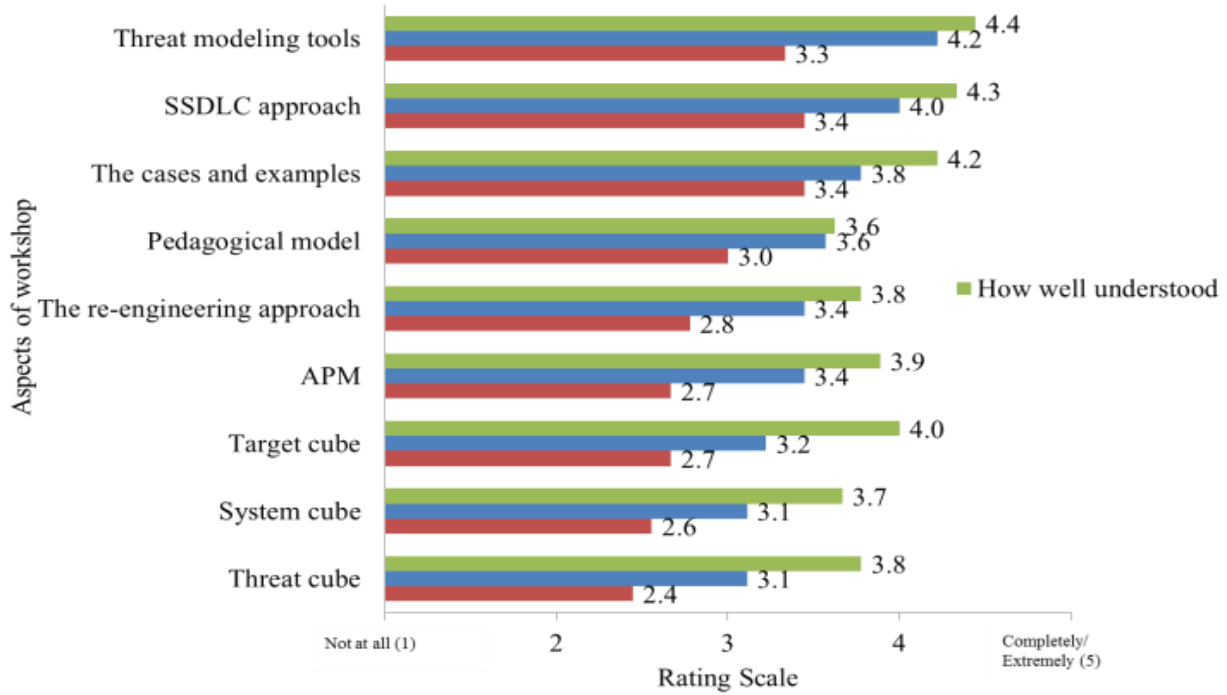


Figure 7: Attitudes and Plans Regarding Workshop Topics

Workshop Component - How Well Understood?

- In all cases, the components were better understood than they were perceived as useful.
- Further, the technical tools were ranked higher in value than the theoretical models.
- The threat modeling tool, the SDL, and the cases were the highest ranked and in that order.
- The software reengineering process was ranked in the middle, while the APM and pedagogical models were least valued.
- The threat modeling tool was also the best understood workshop component.

Workshop Component - How Relevant?

- Participants felt the least able to incorporate the theoretical models and the least convinced of their relevance to their students, and most able to apply **the examples, approaches, and tools, seen as more relevant**.
- **The threat modeling tool** was seen as the most relevant topic covered, with seven of the nine selecting a response on the “relevant” side of the scale—four of them indicating “completely relevant.” The other two selected the scale’s midpoint.

Workshop Component - How Likely to Implement?

- **The SDL approach**:
  - Eight understood it.
  - Six found it relevant, and two rated its relevance below the midpoint.
  - Five are likely to implement it, while three are unlikely.
- **The cases and examples**:
  - Seven understood them.
  - Six found them relevant and the others selected the scale's midpoint.
  - Four are likely to implement them, and four selected the midpoint.

Final Participant Comment

- What components of the workshop we should continue to use in the future?
  - Participants were evenly divided between the practical and the theoretical, with three suggesting we keep the case examples and hands on exercises, and another three suggesting we keep the APM model.
- When asked what to change
  - Three asked for less theory.
  - More opportunity to process the information through more discussion,
  - More group work, and more hands-on exercises.
  - Adding a validation library for each programming language.

Final Participant Comment

- Why it has been hard to teach secure coding in their courses.
  - Two themes emerged: not knowing what to teach in their classes; and not having time to do it.
  - One person also brought up the need to collaborate with other faculty members within his/her program.
- Whether it remains difficult content to teach after exposure in the workshop.
  - Five indicated that it would not be as difficult to teach secure coding concepts, with two adding that the workshop moved their thinking forward on the subject.
  - Two indicated it would be difficult, still, but for different reasons.
- When asked for other comments
  - Learning from hands-on exercises.

Lessons Learned

- **Lack of time** (either in the curriculum itself or in faculty time to develop new course materials or alter existing ones) hindered participants from injecting security topics into their own courses, although they **recognized** that security topics are important.
- Secondly, although participants could easily understand secure coding concepts—like SDL, the APM model, and threat modeling—and **liked the hands-on practices**, they strongly wanted more already **prepared videos and assignment examples that demonstrate the transformation of non-secure to secure code**.

Lessons Learned

- This surprised us since we expected that being able to produce their own assignments from existing legacy assignments would engage them.
- We attributed this lack of interest in our process **to lack of time to prepare their own materials**, even if given an efficient process to convert already created materials.
- We did conclude from their responses that the software reengineering approach using UML modeling was a good method for teaching secure coding concepts, but primarily if the examples are presented **in completed videos**, instead of having the participants apply the method to their own legacy lab assignments to produce exercises themselves.

Lessons Learned

- Based on the participants' responses, we concluded that the Microsoft TMT and SSDL were well-understood.; however, likeliness-to-implement was not as high, again given **time constraints** of the majority of participants.
- We concluded that preparing for the next workshop offering, we must create threat model examples and a variety of different cases in different languages so that they are ready to use by participants, **as opposed to instructing them on how to build their own**.

Lessons Learned

- Given the popularity of the TMT and the SDL, we plan to invite **guest lecturers from industry** who have used these tools to present how these tools have assisted them in improving the security of their products. We expect that this added emphasis may encourage instructors to use these tools in their classes.
- The tepid response to using the re-engineering approach was not entirely surprising, since many participants are not familiar with UML diagrams; however when presented **with videos of a completed reengineering process**, including narrative, the approach made more sense.

Lessons Learned

- Participants challenged whether the cases were sufficiently "**real life**."
  - To improve their relevance, we will tie our cases more directly to the NVD and CWE.
- The APM model was seen as a reasonable way to position **the subject of secure code within the information assurance body of knowledge**, although participants were vague about how to use the model in their classes.

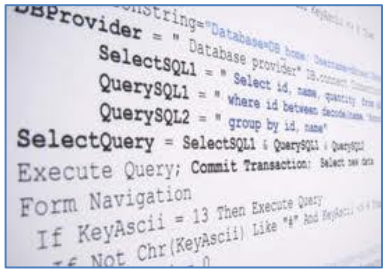
Future Work

- Integrating Security into Your Current Courses with Minimal Effort
- More clear integration between the re-engineering process and each case study.

Available Instructional Media for Your Classes

- Video clips

- Misuse cases
- Programming assignments



- Lecture slides
- Standard repository resources

Question and Answer