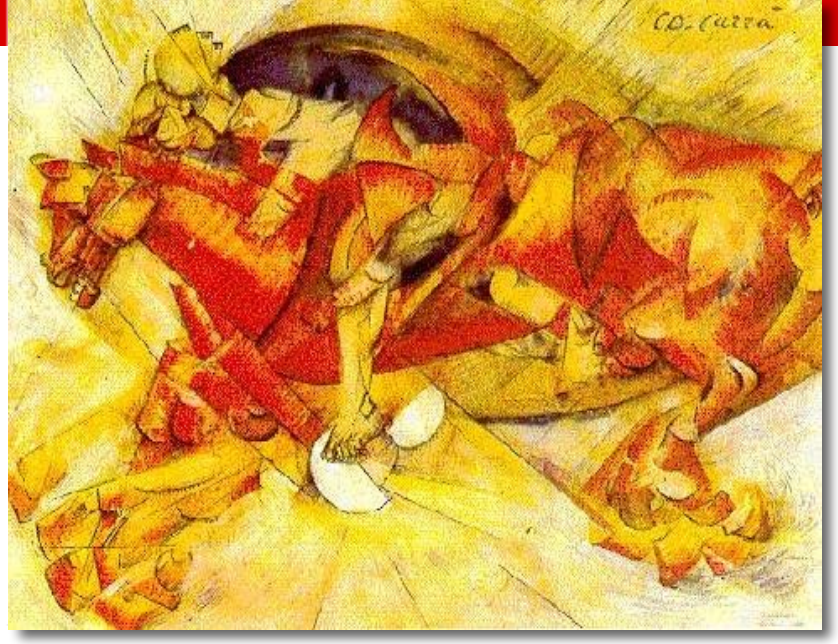


Faster Evaluations: A Matter of Timing

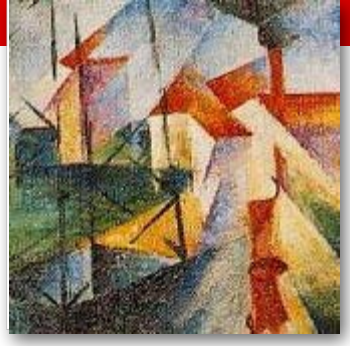
Courtney Cavness, CISSP
atsec information security corporation

The need for speed

What makes evaluations go faster?

- Developer having adequate security procedures?
- Developer having sufficient evidence?
- Perfect code?
- Experienced evaluators?
- Knowledgeable certifiers?
- *Enforced* timelines?

Faster evaluations require **communication** and **collaboration**.



CCMB vision statement

The vision statement shortens timelines by:

- Reducing the effort by labs
  - ✓Limited EALs
- Promoting repeatability among developers
  - ✓Required conformance to existing PPs

What is a reasonable timeline?

Scheme – 3 months to certify lab reports

Labs – variable depending on the state of evidence

Developers – one or more years to produce satisfactory and complete evidence

End users – as soon as the product is made available

Developer Timelines: Why so long?

- Dependent on another group or 3rd party
- Inexperienced with level of effort necessary to generate non-existent or sufficient evidence
- Inadequate process in place
- Code change necessary (vulnerability discovered/new feature required)
- Economic climate

  Unexpected layoffs resulting in lack of manpower and expertise required to generate evidence/respond to feedback

Can lead to developers providing “best case scenario” scheduling estimates

Results in:

- ☹ **Stagnation of evidence development**
- ☹ **Multiple deadline extension requests**
- ☹ **Unmet expectations by both labs and schemes**
- ☹ **Scheduling conflicts**
- ☹ **Delayed feedback**

Loss of confidence in all parties

Suggested solution…

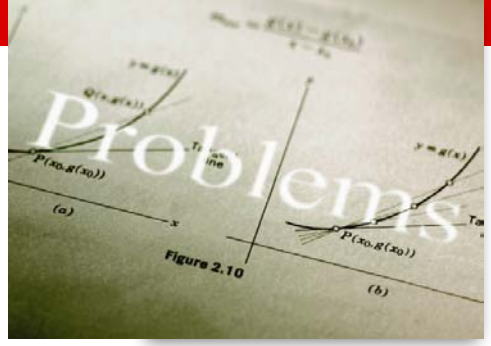
Perform evaluation *after* the product has GA'ed

During development, the developer's focus is on producing the TOE; not on producing evaluation evidence.

- Flaw fixes
- Changing feature specifications
- Unexpected business disruptions

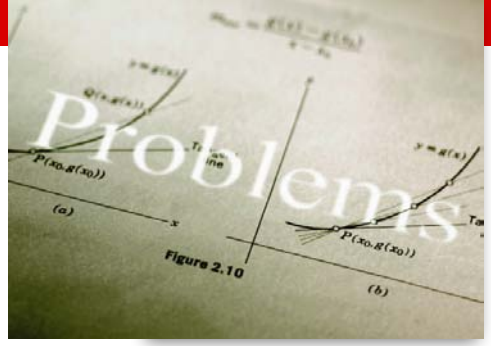
…causing "extra" efforts (like certification) to take a back seat to getting the product out the door.

Problems with postponing evaluation until GA

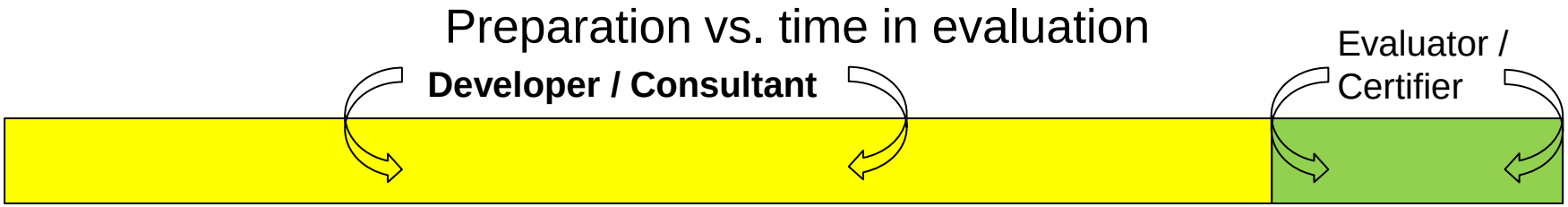


- Evaluator may find problems *before* the developer / certifier / end user
  - In code
  - Security functionality
  - Compliance to PPs
- Problems found later in the development cycle are more expensive to fix
- End users need the evaluated TOE as soon as possible after GA
- Higher risk of getting a successful certification ***after*** the product has been discontinued

“How long it takes”



- Is most variable for developers (front end work)
- Other delays:
  - Scheme formalities/interpretations (i.e., random number generator and entropy requirements)
  - Scheme Limitations (i.e., scheme-acceptable PPs)
  - Product complexity
  - Resources (all parties)
  - Resolution to project-specific issues (i.e., what to test, how to test, site visit requirements)



Redefine the timing

1. Specify when the clock starts ticking

The “clock” should start at different times for different stakeholders:

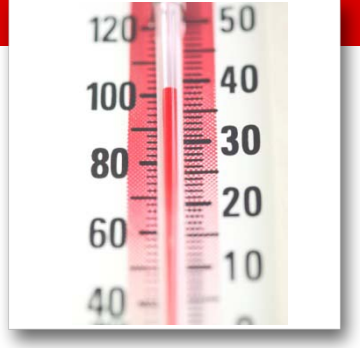
- Labs could begin consulting/evaluation efforts during the TOE’s development, providing valuable input regarding compliance to PPs and early identification of potential vulnerabilities and evidence gaps.
- Meanwhile, schemes could save time/effort by not officially assigning manpower to a project until the developer’s consulting effort is complete and evidence has been provided to the lab.
  - Getting reports all at once (or in quick succession) may change how schemes approach developer contact/feedback during evaluation (i.e., VORs, kick-off meetings, site visit attendance)

Redefine the timing (cont'd)

2. Create a "Freeze process" to stop the ticking clock

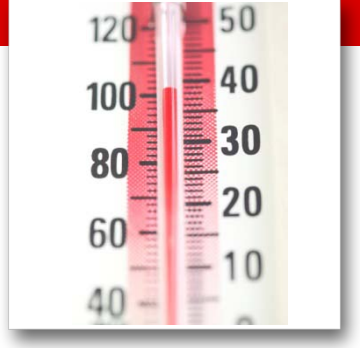
- Not punitive (a "kick you out" mindset)
- Remove the stigma of delays
- Promotes proactive developer response to major slowdowns
- Promotes communication and collaboration

Freeze Process Defined



- Can occur at any point after evaluation begins.
- Provides a clear path to re-instate evaluation effort once the business crisis has passed.
- Formalized by a process understood by all parties (i.e., a form submitted to the scheme by either the developer or lab on developer's behalf).
- May include a payment point to provide compensation for work done to date by all parties (in case evaluation does not resume).

Freeze Process Defined (cont'd)



- Can be for any reason. Can be unspecified.
- Limited to one after the scheme begins evaluation effort (or else incur penalty fee).
- Should have an expiration date – not linger indefinitely.
- Applicable scheme lists of "Products in Evaluation" which serve as a business/sales incentive, should have an "on hold" designation for frozen evaluations.

  Note: End users themselves may place purchasing restrictions on those listings.

Conclusion

- It is ineffective for all parties involved to begin work at the same time due to the dependency on the developer to first provide adequate and sufficient evidence.
- Business doesn’t always run on a predictable schedule. Evaluation timelines need some defined flexibility such as a “freeze process.”
- Evaluation effort requires multiple parties with different expectations working together. Therefore, communication is key to preventing misconceptions and loss of confidence when schedules slip.
- Unique, individual variables prevent specifying “how long an evaluation takes.” But delaying the involvement of other parties will minimize their timeframe.