OWASP Europe Conference 2008

Security framework is not in the code

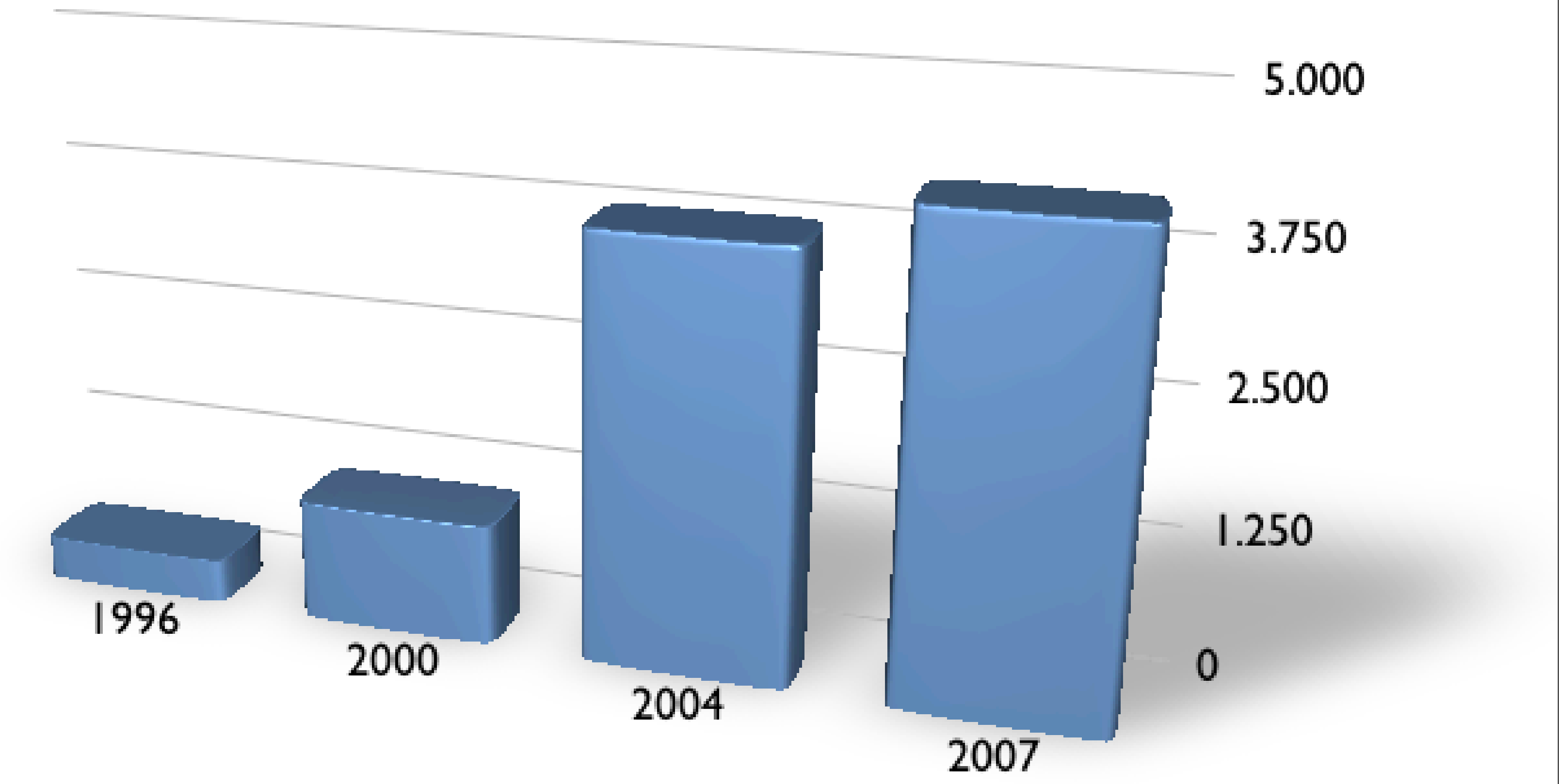
Sam Reghenzi

OWASP

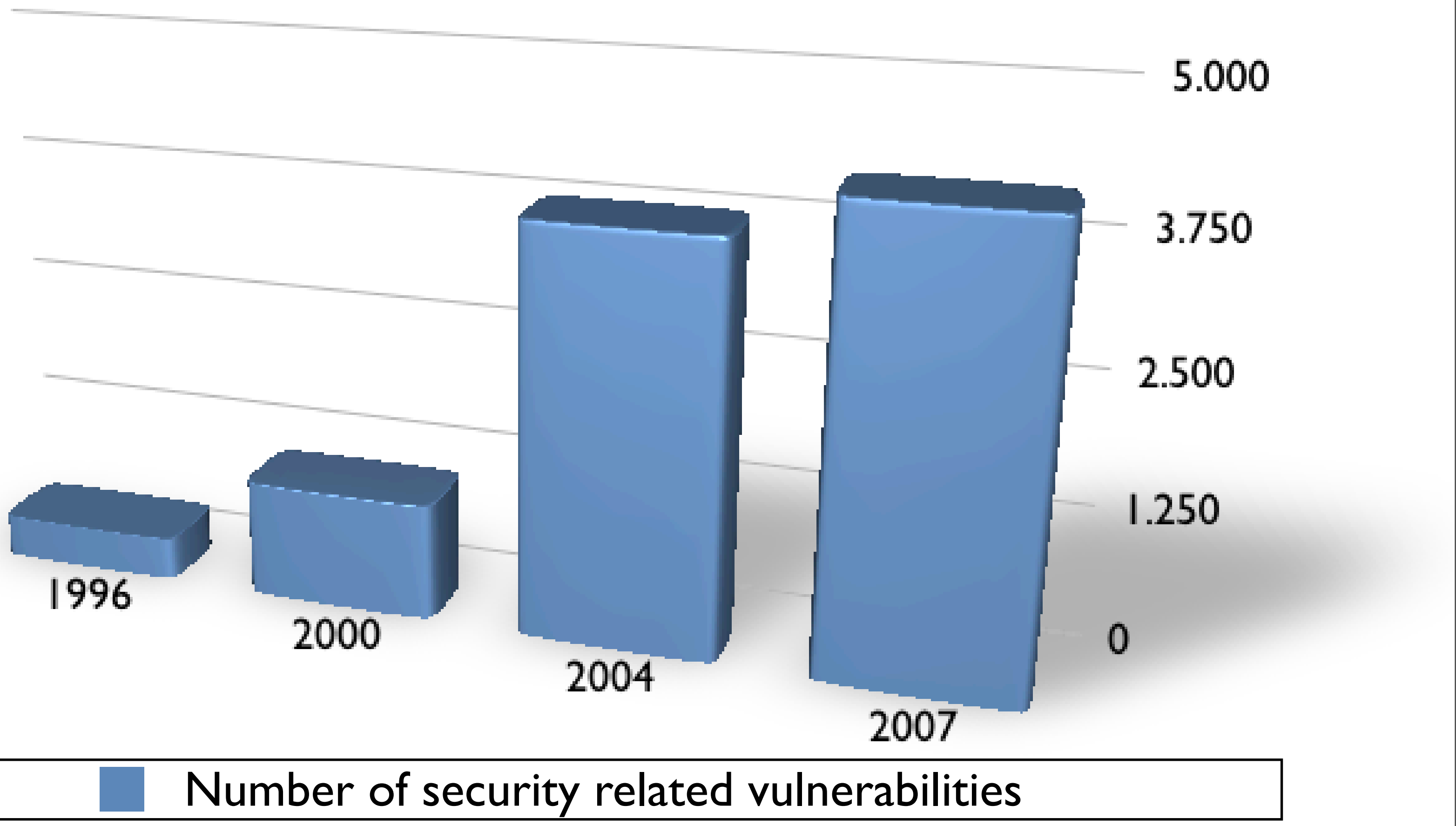
The OWASP Foundation
http://www.owasp.org

Do we really need more security in our software?

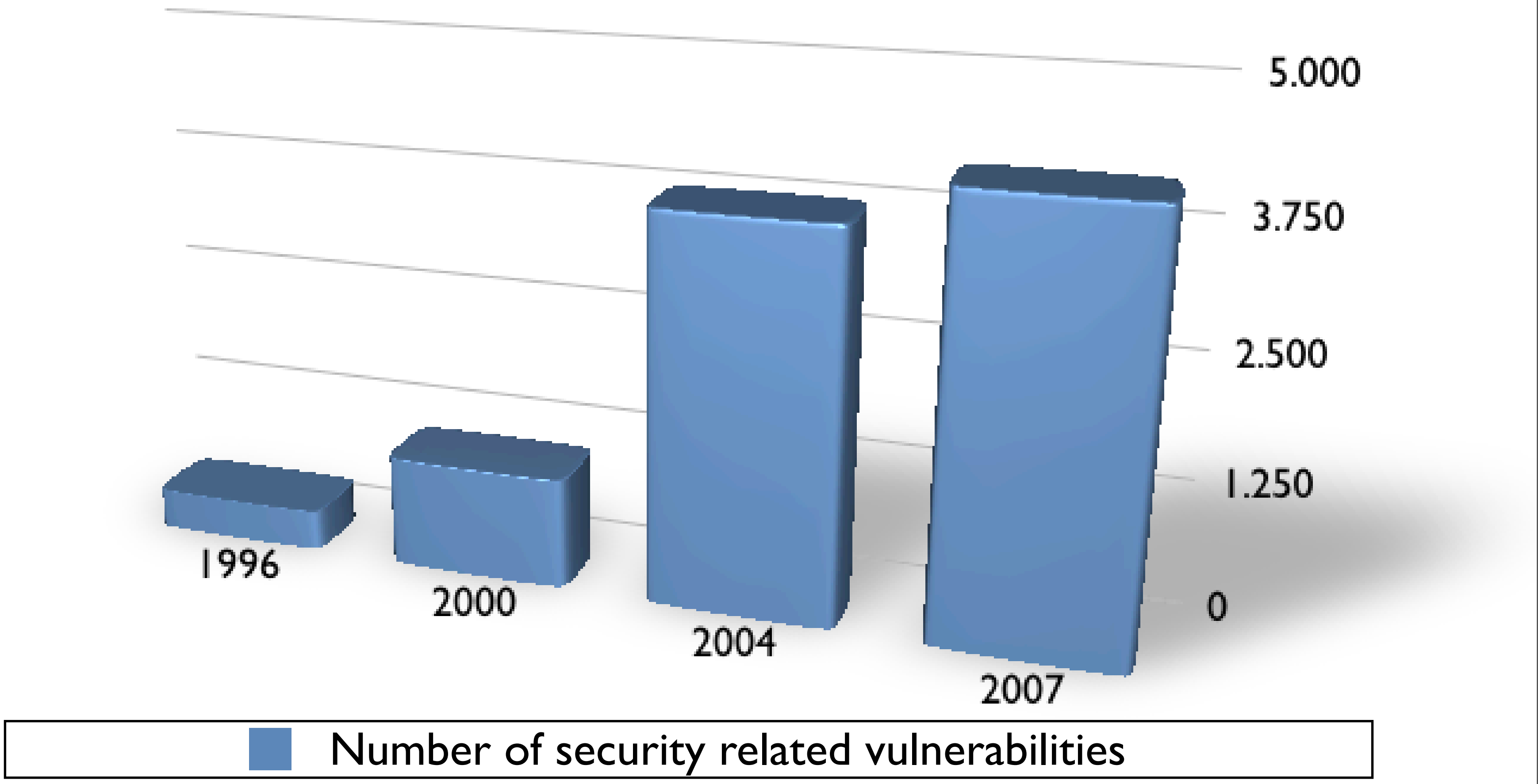
Do we really need more security in our software?



Do we really need more security in our software?



Do we really need more security in our software?



We need to build better software

What we mean with Security Framework

It is not

- Authentication and authorization
- Encryption
- Firewall software

It could be

- An enterprise security approach
- A risk management framework for security related threats
- Defined steps in your (Secure) development life cycle

What we mean with Security Framework

It is not

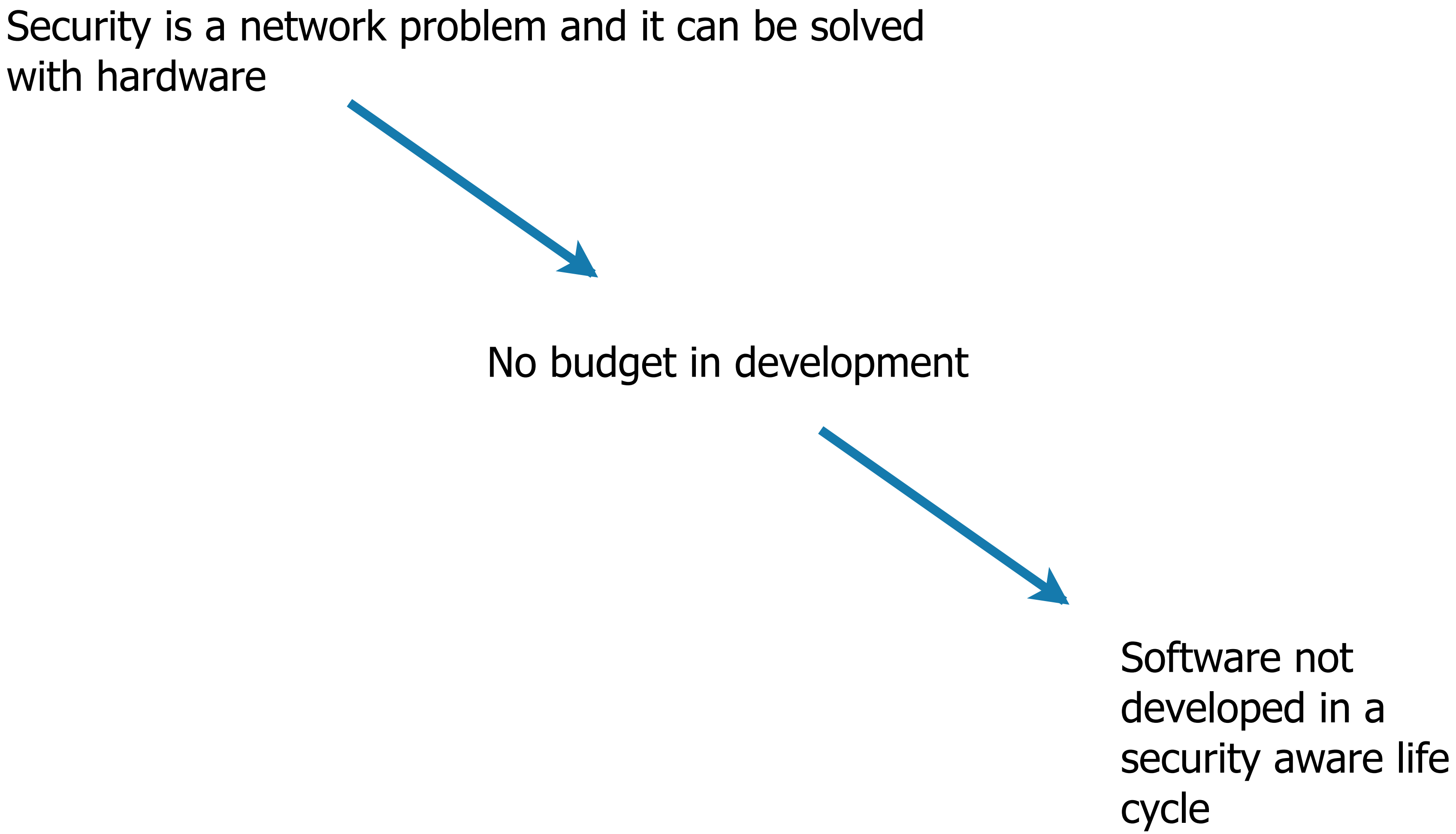
- Authentication and authorization
- Encryption
- Firewall software

It could be

- An enterprise security approach
- A risk management framework for security related threats
- Defined steps in your (Secure) development life cycle

Application security is inside the application not around it

Traditions (And other bad habits)



Security is a network problem and it can be solved with hardware

No budget in development

Software not developed in a security aware life cycle

Establish security in your DL

Software engineering

- Find best practice to fit your team or company
- Test for abuse, not only for good use
- Measure code, bug and progress

Social engineering

- Make good friends
- Be aware of your business compliancy
- Wait... something bad will happen

The ROI Problem

Security in software development brings no direct revenue

#1 **Reduce costs**

#2 **Bring evidence of risks**

#3 **Sell security as a value**

[Static]Code analysis

- Add security awareness in code reviews
- Add security blue prints in automatic code analysis
- Fix codebase and third party software

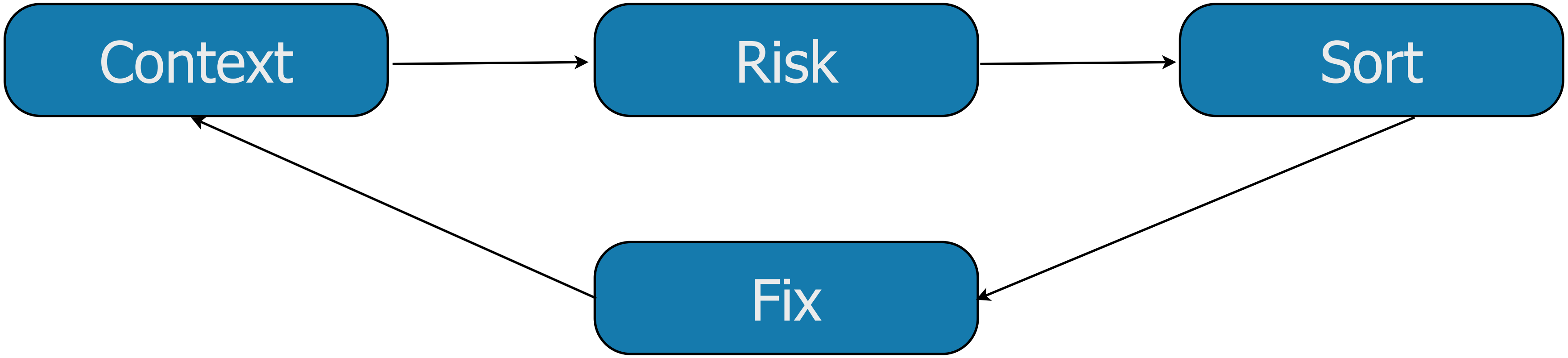
[Static]Code analysis

The poor man software security

- Add security awareness in code reviews
- Add security blue prints in automatic code analysis
- Fix codebase and third party software

Security Risk management

Manage knowledge, identify risks, rank them and fix them



Security Risk management

#1

- Gather documentation
- Gather information from management
- Gather information from the team
- Gather information from artifacts

#2 **Organize everything**

#3 **Make the deal**

Hot stages of SDLC

- The architectural design
- The development
- The test
- The enhancement

- User stories
- Test driven
- Iterations

- Code review
- Abuse cases
- Penetration testing
- Security requirements
- Risk analysis

Hot stages of SDLC

Traditional

- The architectural design
- The development
- The test
- The enhancement

- User stories
- Test driven
- Iterations

- Code review
- Abuse cases
- Penetration testing
- Security requirements
- Risk analysis

Hot stages of SDLC

Traditional

- The architectural design
- The development
- The test
- The enhancement

Agile

- User stories
- Test driven
- Iterations

- Code review
- Abuse cases
- Penetration testing
- Security requirements
- Risk analysis

Hot stages of SDLC

Traditional

- The architectural design
- The development
- The test
- The enhancement

Agile

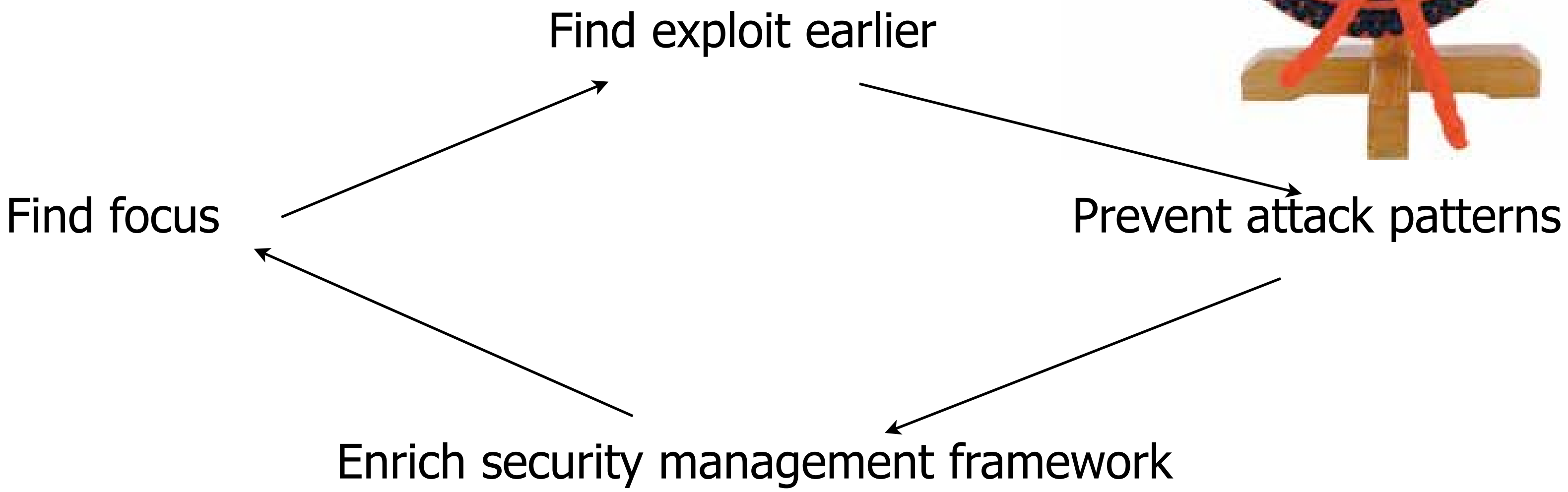
- User stories
- Test driven
- Iterations

Touchpoints

- Code review
- Abuse cases
- Penetration testing
- Security requirements
- Risk analysis

Historical knowledge

Know your enemies



Find exploit earlier

Prevent attack patterns

Enrich security management framework

Find focus

Tips

- Jump on the **High availability** train
- Mitigate Web 2.0
- Deliver something concrete
- In Rome act like a Roman

Q&A

?