IBM SECURITY DAY 2011

Innovare con sicurezza per aprire al futuro

Simone Riccetti

Applicazioni web:Security by design

IBM

Perchè il problema continua a crescere?

Connettività:

Internet
L'incremento del numero e delle tipologie d vettori di attacco è proporzionale all'incremento di connettività.

SOA/Web Services
Applicazioni legacy non progettate per essere in rete, sono visibili come servizi.

Sistemi Legacy
Non sempre supportano le moderne funzionalità di sicurezza (es. Autenticazione)

Estensibilità:

Il software è sempre più "estensibile" , es browser plug-in, dynamic loadable device driver

L'estensibilità dei software rende difficile prevederne la superficie di attacco nel tempo

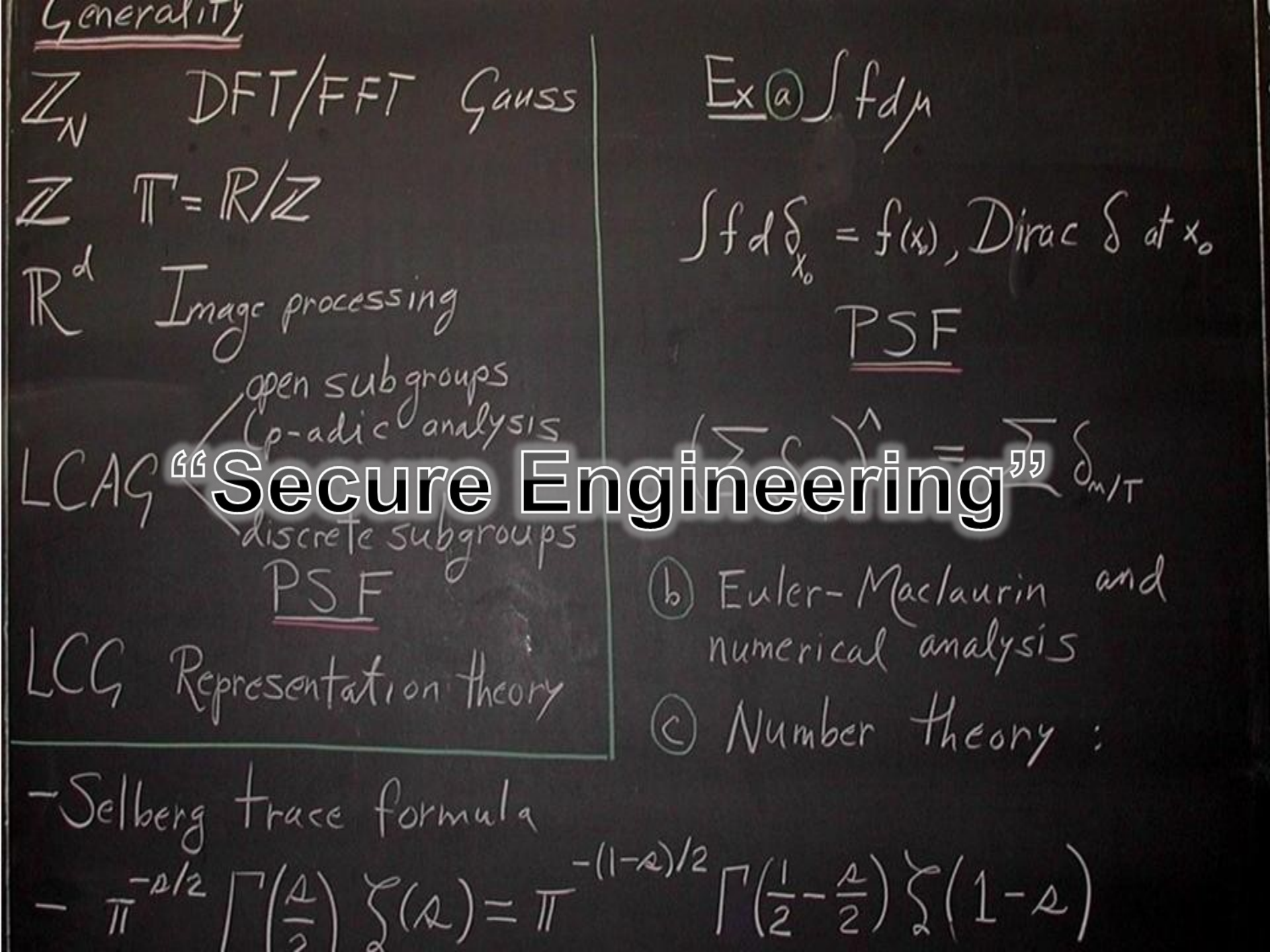
Complessità:

La complessità delle architetture software è sempre maggiore

Nel 1990, Windows 3.1 aveva 2,500,000 linee di codice.

Windows XP ha circa 40,000,000 linee di codice.

Il numero di bugs è proporzionale al numero di linee di codice



“Secure Engineering”

Secure Engineering

- Una serie di concetti, principi e best practice con l'obiettivo di integrare la sicurezza nel software progettato.
- Riguarda tutte le fasi del ciclo di sviluppo del software

Security by Design

Security by Implementation

Security by Deployment

Secure Engineering non riguarda solo la fase di implementazione, ma è un approccio end-to-end

Quanto costa correggere una vulnerabilità?

80% dei costi di sviluppo sono spesi per identificare e correggere gli errori

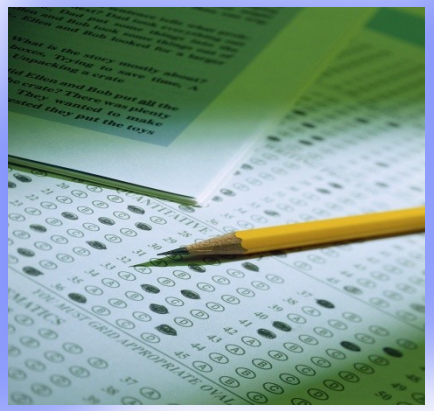
National Institute of Standards & Technology

Durante la fase di coding

$25/difetto

Durante la fase di build

$100/difetto



Durante la fase di test

$450/difetto

Quando il prodotto è stato rilasciato

$16,000/difetto

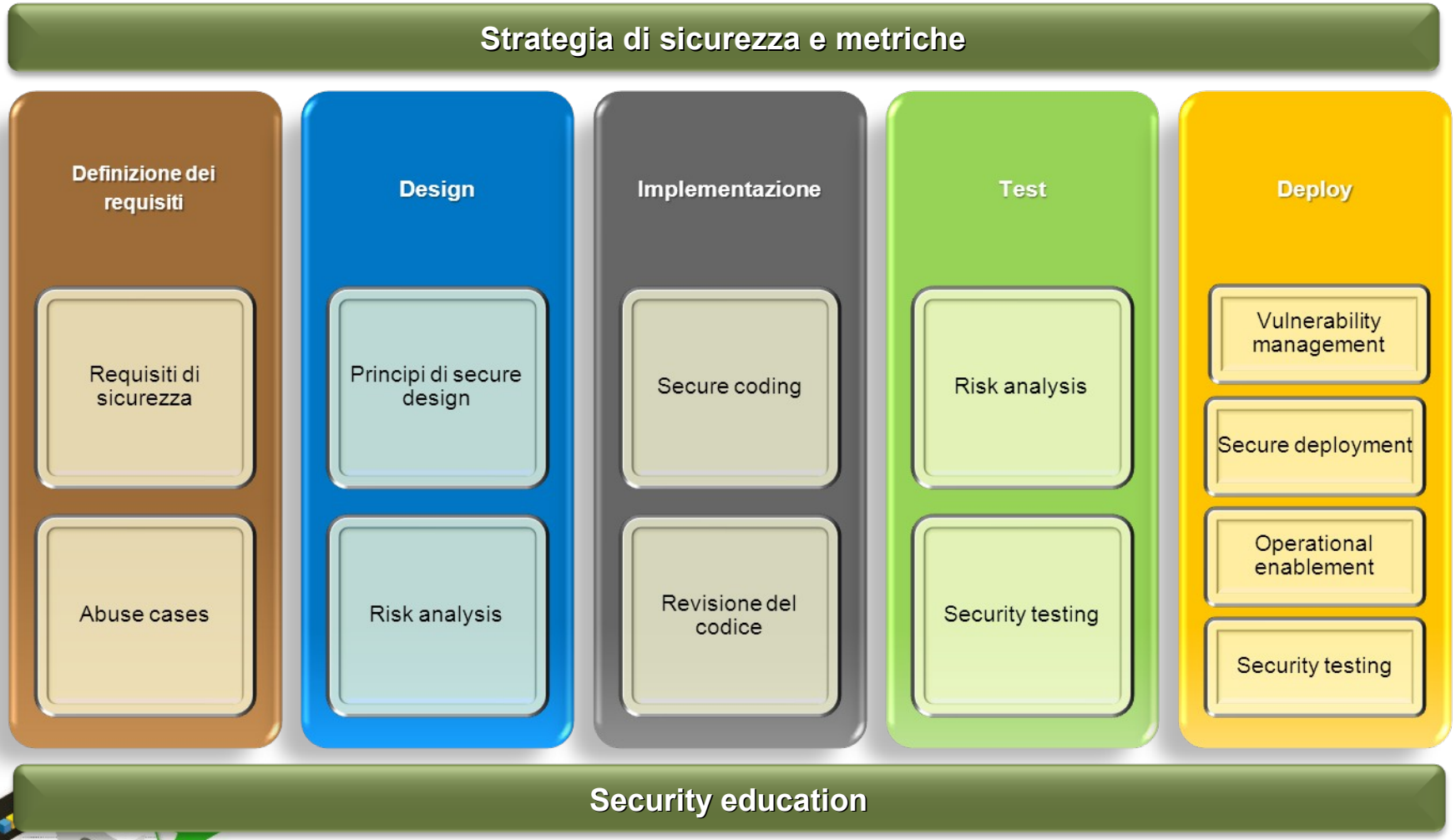
+

Perdita della fiducia del cliente, danni di immagine, etc.

Caper Jones, Applied Software Measurement, 1996

** Source: 2008 GBS Industry standard study. Defect cost derived in assuming it takes 8 hrs to find, fix and repair a defect when found in code and unit test. Defect FFR cost for other phases calculated by using the multiplier on a blended rate of $80/hr.*

Security Engineering & Software Development Life Cycle

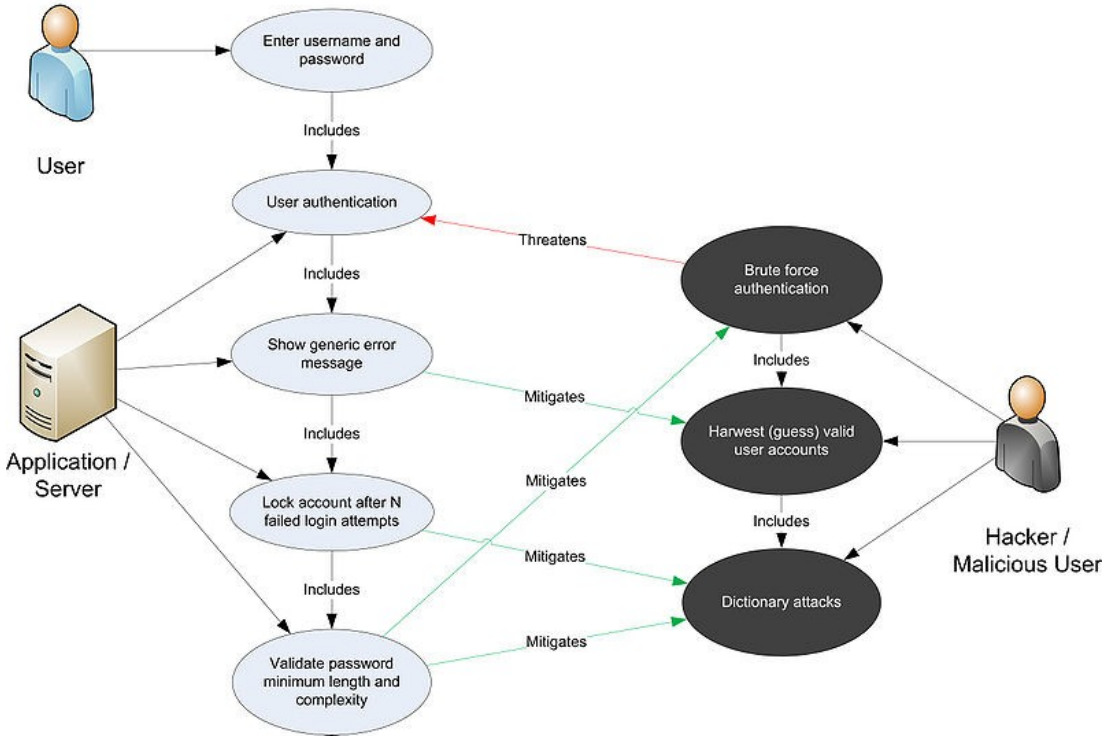


Quali sono le cause principali?

1. Requisiti di sicurezza non definiti o poco chiari
2. Implementazione errata di requisiti corretti
3. Nella fase di deployment e configurazione non rispecchia i requisiti e le modalità operative previste

Use Cases e Misuse Cases

- Oltre agli *Use Cases* anche i *Misuse Cases*



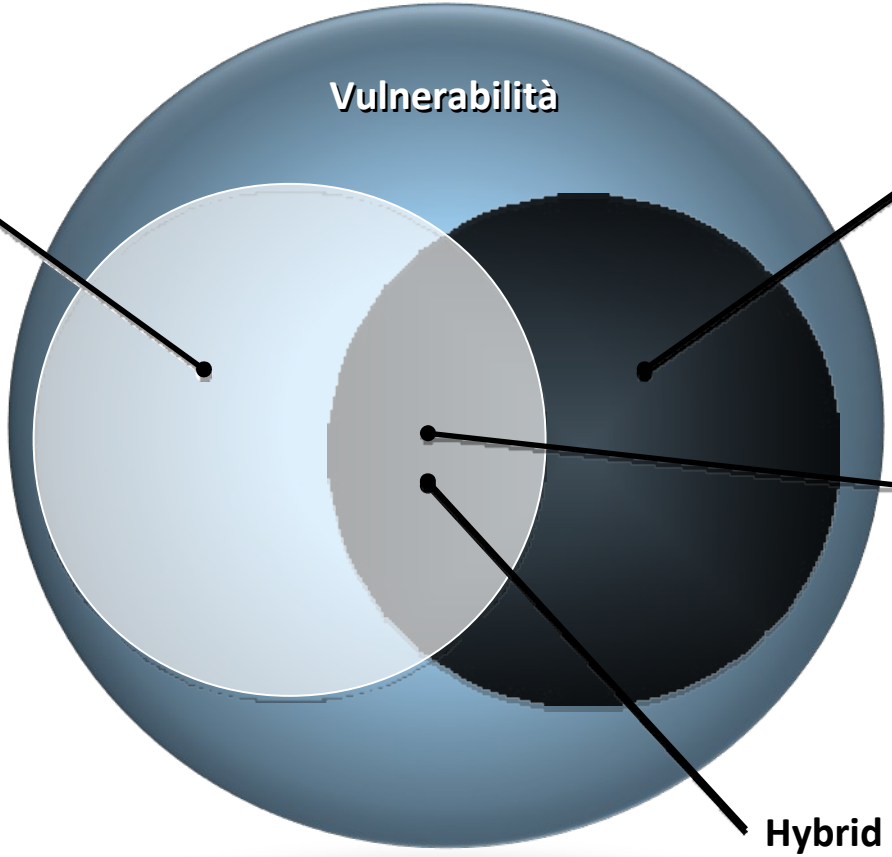
E' importante avere la prospettiva dell'attacker

Rer: http://www.owasp.org/index.php/Testing_Guide_Introduction

Es. Taint String

```
// ...
  String username = request.getParameter("username");
  String password = request.getParameter("password");
  // ...
  String query = "SELECT * from tUsers where " +
    "userid='" + username + "' " + "AND password='" + password +
"'";
  // ...
  ResultSet rs = stmt.executeQuery(query);
```

Combinazione dei test di sicurezza



Static Analysis

- Null pointer dereference
- Threading issues
- Code quality issues
- Issues in dead code
- Insecure crypto functions
- Issues in back-end application code
- Stored vulnerabilities
- Complex SQL injection

Dynamic Analysis

- Environment configuration issues
- Patch level issues
- Runtime privileges issues
- Authentication issues
- Protocol parser/serializer issues
- Session management issues
- Issues in external 3rd party web components
- Cross-site request forgery
- Malware analysis

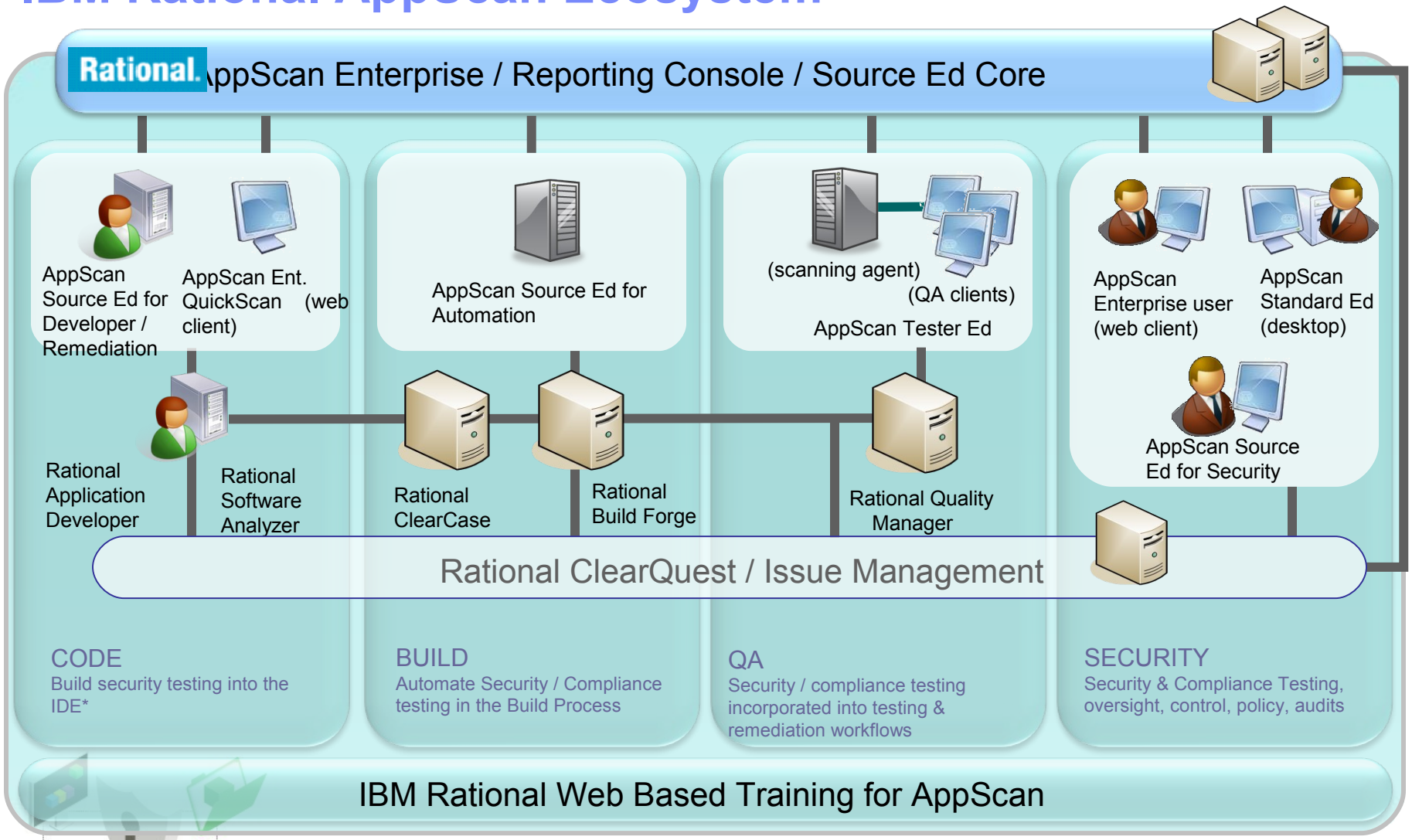
Dynamic & Static

- SQL Injection
- Cross Site Scripting
- HTTP Response Splitting
- OS Commanding
- LDAP Injection
- XPath Injection
- Path Traversal
- Buffer Overflows
- Format String Issues

Hybrid Analysis*

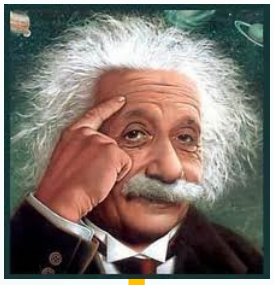
- Client-side issues
  - DOM-based XSS
  - DOM-based Open Redirect
  - HTML5 Notification Phishing
  - HTML5 Client-side SQL Injection
  - Etc.

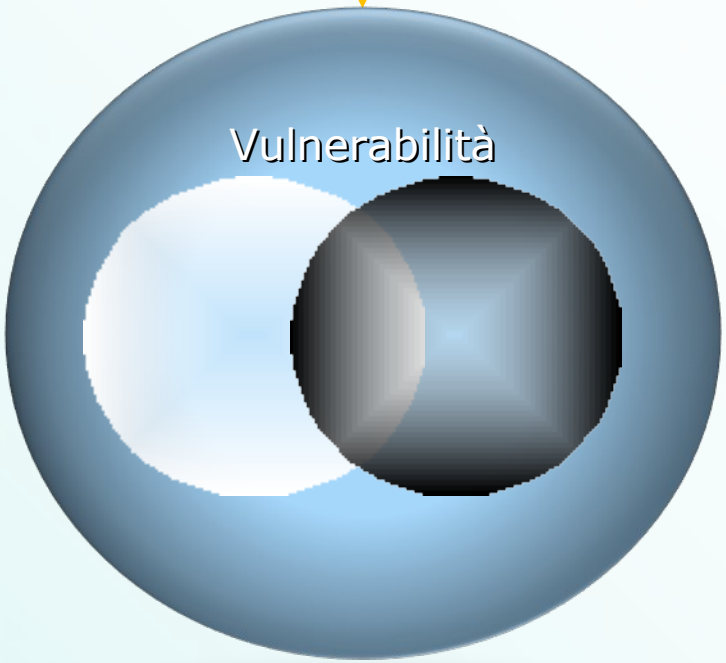
IBM Rational AppScan Ecosystem



Non solo i tool…

Security Architecture
Security Implentation
Code review (manuale)
……





Grazie!

simone.riccetti@it.ibm.com