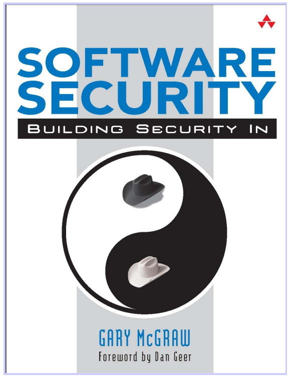
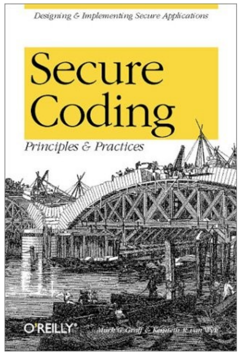
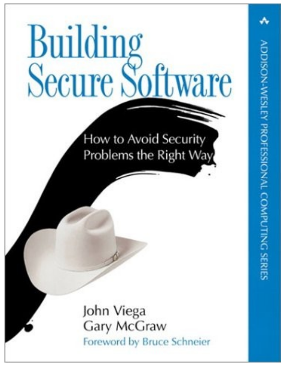
KRvW
Associates

Introduction

SepAppDev 2007







cigital KRvW Associates

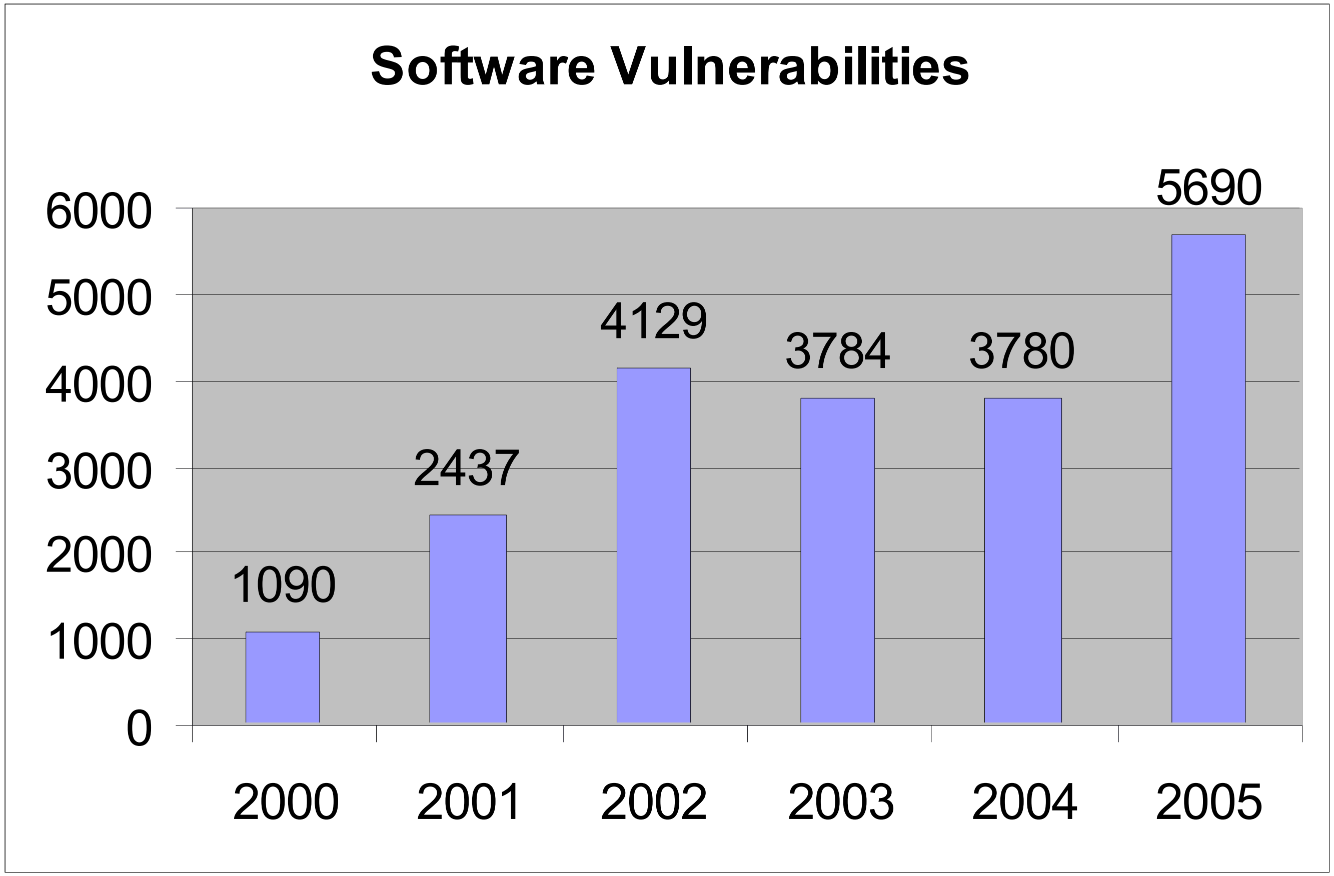
Contents of the Course

- Not so much in chronological order, but
  - Security objectives
  - Development process
  - Mechanisms in current technologies
  - Design
  - Coding
  - Quality assurance

The Problem

KRvW Associates

Software vulnerability growth



The Trinity Of Trouble: Connectivity

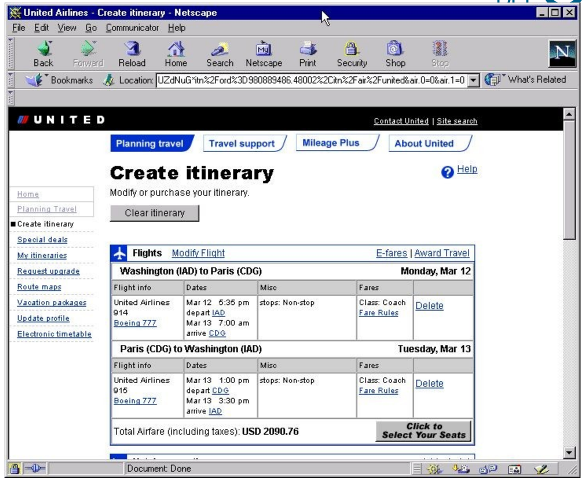
- The Internet is everywhere and most of our software is on it
- When was the last time that you did business with a major vendor who had no Internet connectivity?
- Tried VoIP on your mobile phone in a coffee shop WiFi hotspot yet?

The network is the computer.

Sun microsystems

The Trinity Of Trouble: DEL Complexity

- A simple user interface can be enormously complex “under the hood”
- Consider what happens behind the scenes in one of today’s AJAX web applications
- But it sure does make for a compelling “user experience”


United Airlines - Create itinerary - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: UZdNuG*itn%2Ford%3D980889486.48002%2Citn%2Fair%2Funited&air.0=0&air.1=0 What's Related
UNITED
Contact United | Site search
Planning travel
Travel support
Mileage Plus
About United
Home
Planning Travel
Create itinerary
Special deals
My itineraries
Request upgrade
Route maps
Vacation packages
Update profile
Electronic timetable
Help
Clear itinerary
Click to Select Your Seats
Document: Done

Create itinerary

Modify or purchase your itinerary.

Flights Modify Flight E-fares | Award Travel

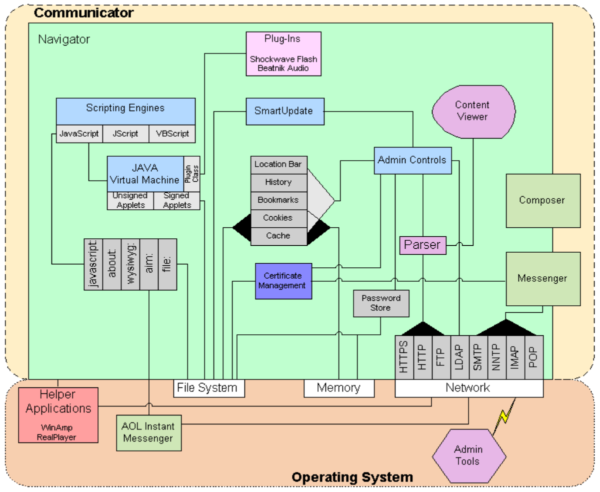
Washington (IAD) to Paris (CDG) **Monday, Mar 12**

Flight info	Dates	Misc	Fares	
United Airlines 914 Boeing 777	Mar 12 5:35 pm depart IAD Mar 13 7:00 am arrive CDG	stops: Non-stop	Class: Coach Fare Rules	Delete

Paris (CDG) to Washington (IAD) **Tuesday, Mar 13**

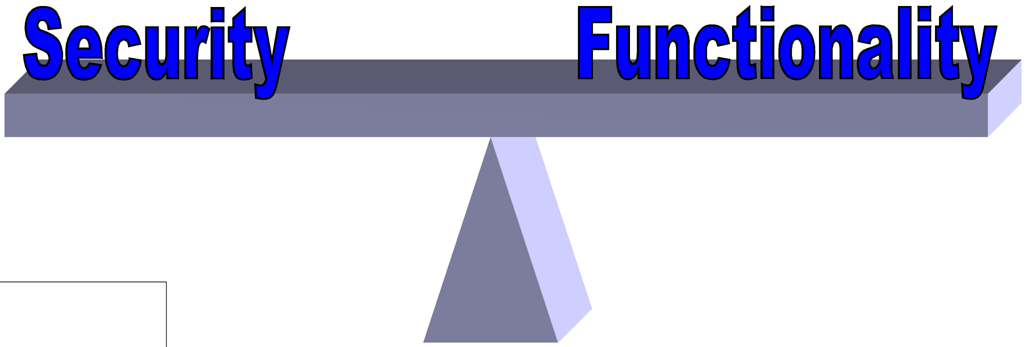
Flight info	Dates	Misc	Fares	
United Airlines 915 Boeing 777	Mar 13 1:00 pm depart CDG Mar 13 3:30 pm arrive IAD	stops: Non-stop	Class: Coach Fare Rules	Delete

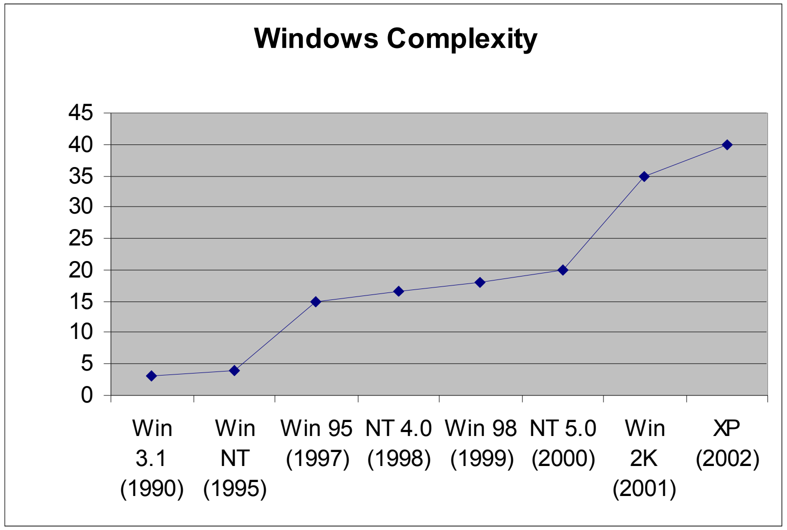
Total Airfare (including taxes): **USD 2090.76**



KRvW Associates

The classic security tradeoff





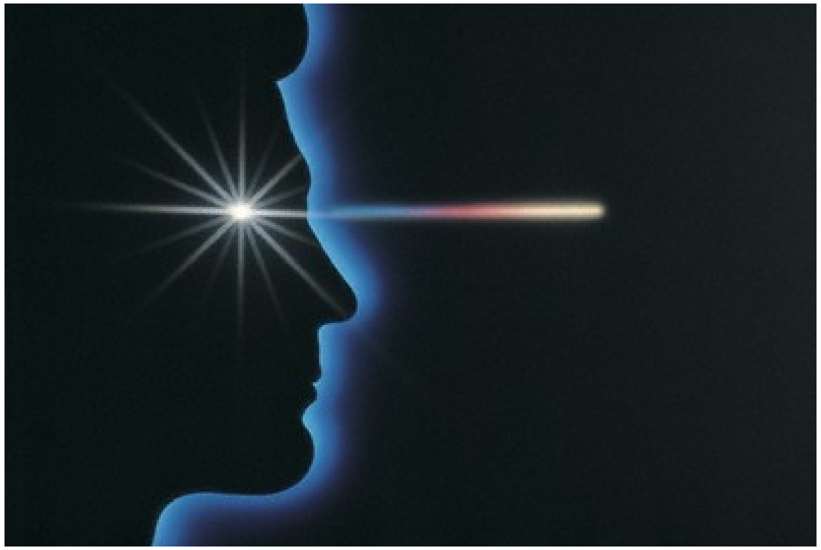
Learning from history

- We don’t pay enough attention to our failures
- Consider other engineering disciplines
  - Transportation
  - Construction
  - Medical

cigital KRvW Associates

Focus on function

- Too much attention is paid to functional spec
- Consider what can go wrong as well



NO SIGNAL

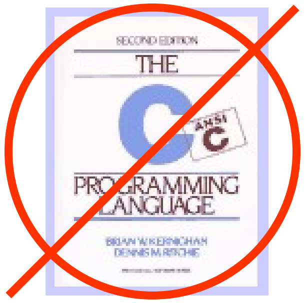
Security problems are complicated

CODE

- Buffer overflow
  - String format
  - One-stage attacks
- Race conditions
  - TOCTOU (time of check to time of use)
- Unsafe environment variables
- Unsafe system calls
  - System()
- Untrusted input problems

DESIGN

- Misuse of software “feature”
- Flawed cryptographic key management
- Compartmentalization problems in design
- Catastrophic security failure (fragility)
- Insecure or insufficient auditing
- Broken or illogical access control (RBAC over tiers)
- Signing too much code



KRvW Associates

cigital

Code example: The dreaded buffer overflow

- Overwriting the bounds of data objects
- Allocate some bytes, but the language doesn't care if you try to use more

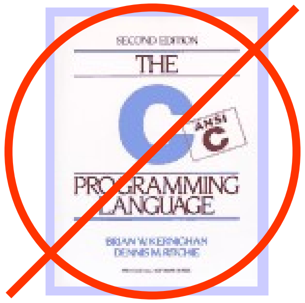
  -
    ```
    char x[12];
    x[12] = '\0';
    ```
- Why was this done? Efficiency!
- Two main flavors of buffers
  - Heap allocated buffers
  - Stack allocated buffers
  - Smashing the stack is the most common attack

- The *second* most pervasive security problem today in terms of reported bugs
- Any guesses what problem has overtaken it recently?

Pervasive C problems

```
void main() {
   char buf[1024];
   gets(buf);
}
```

- How not to get input
  - Attacker can send an infinite string!
  - Chapter 7 of K&R (page 164)



- Calls to watch out for

Instead of:	
gets(buf)	
strcpy(dst, src)	
strcat(dst, src)	
sprintf(buf, fmt, a1,É)	
*scanf(É)	

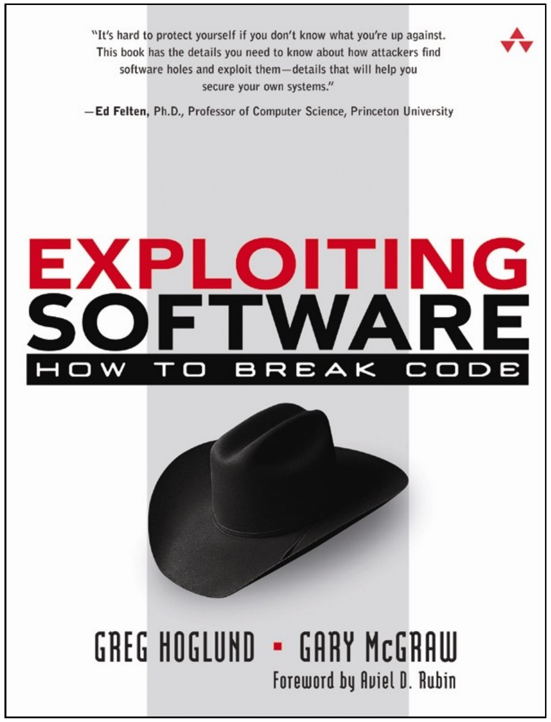
- Hundreds of such calls
- Use static analysis to find these problems
  - ITS4, Fortify
- Careful code review is necessary

Design example: Microsoft WMF

- Windows Metafile Format -- used for interchange of data between programs
  - Design feature included ability to include arbitrary executable data along with a WMF file
  - Feature was included to allow cancellation of print files
  - Attacker could send a WMF file with embedded arbitrary executable code

KRvW Associates

Breaking stuff is important


"It's hard to protect yourself if you don't know what you're up against. This book has the details you need to know about how attackers find software holes and exploit them—details that will help you secure your own systems."

—Ed Felten, Ph.D., Professor of Computer Science, Princeton University

EXPLOITING SOFTWARE

HOW TO BREAK CODE

GREG HOGLUND • GARY McGRAW

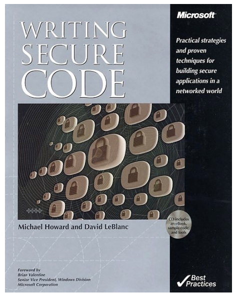
Foreword by Aviel D. Rubin

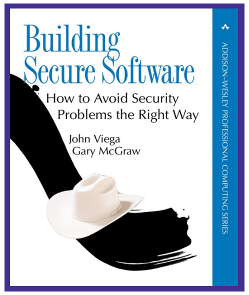
- Learning how to think like an attacker is essential
- Do not shy away from carrying out attacks on your own stuff
  - Engineers learn from stories of failure
- Attacking is fun! Fun is good!

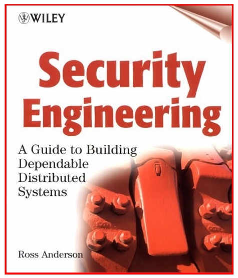
Solutions

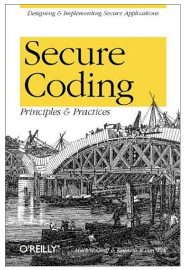
Software security: state of the practice

- *Software security* still in infancy
  - Lacking standards
  - Many "best practices" to choose from
  - Most have yet to really prove themselves
- Information/guidance resources are appearing quickly
  - Study and adopt to your needs
- Tools are getting better, but only cover coding defects
  - Leave much to be done manually

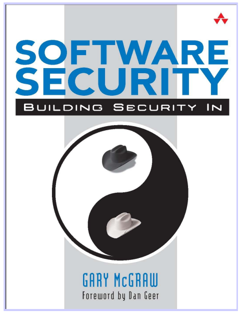








Software security is not security software!
Software security is about building things properly.

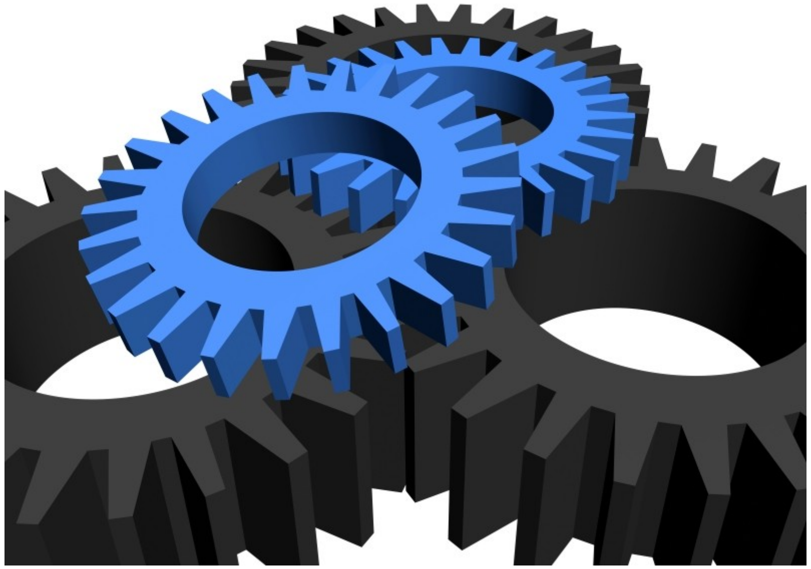


What can be done?

Strive for the following criteria

- Repeatable
- Predictable
- Businesslike
- High quality
- Measurable

Must be firmly embedded into entire existing dev process without breaking it.



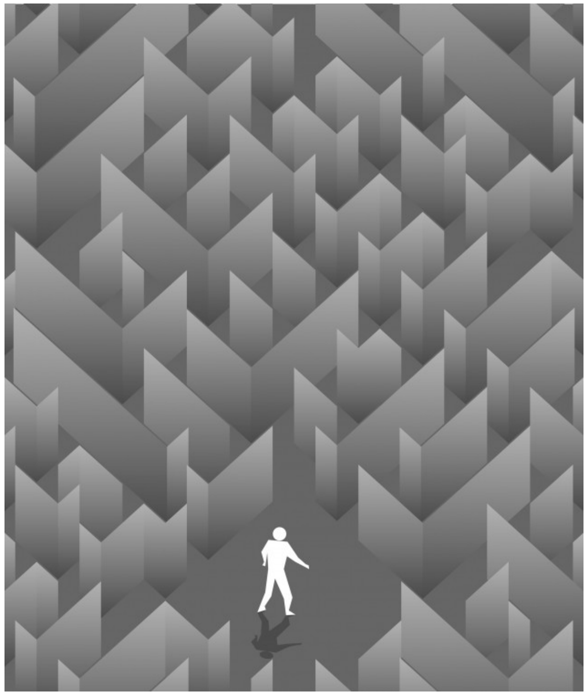
Solution sets abound

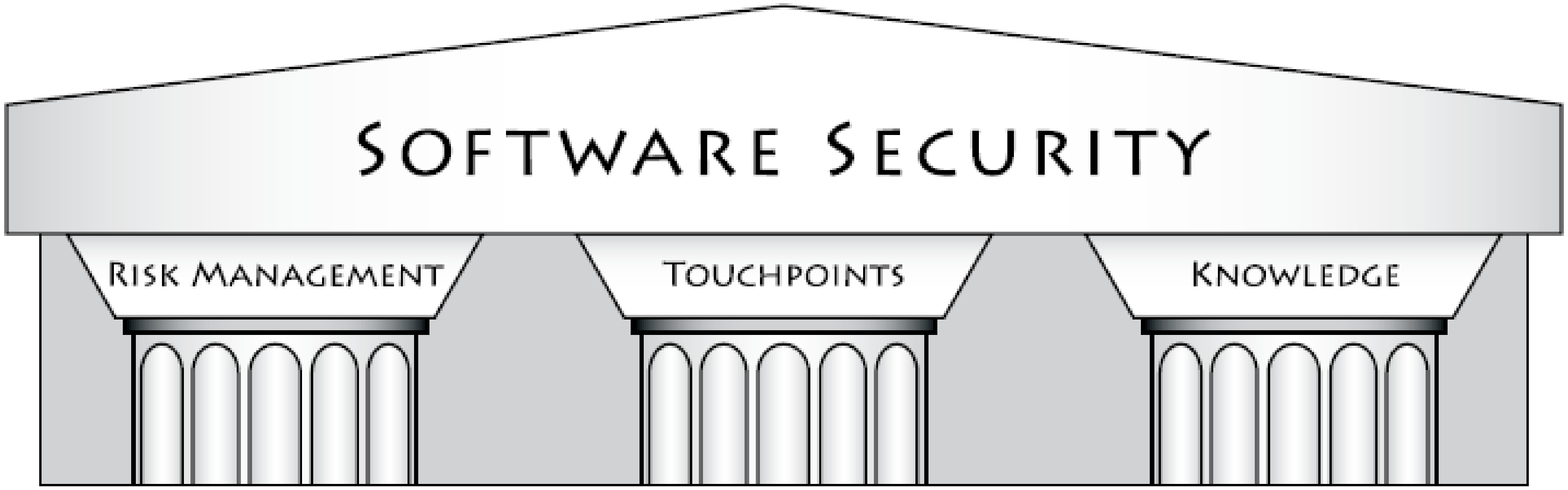
Several "best practices" options to choose from, including

- OWASP's CLASP
- Microsoft's SDL
- Cigital's "touchpoints"

Each has strengths and weaknesses

- Best bet is to learn each and adapt the aspects that work best in your organization
- Alignment with extant build process is vital





Three pillars of software security

- Risk management framework
- Secure SDLC practices or “touchpoints”
- Knowledge catalog

cigital KRvW Associates

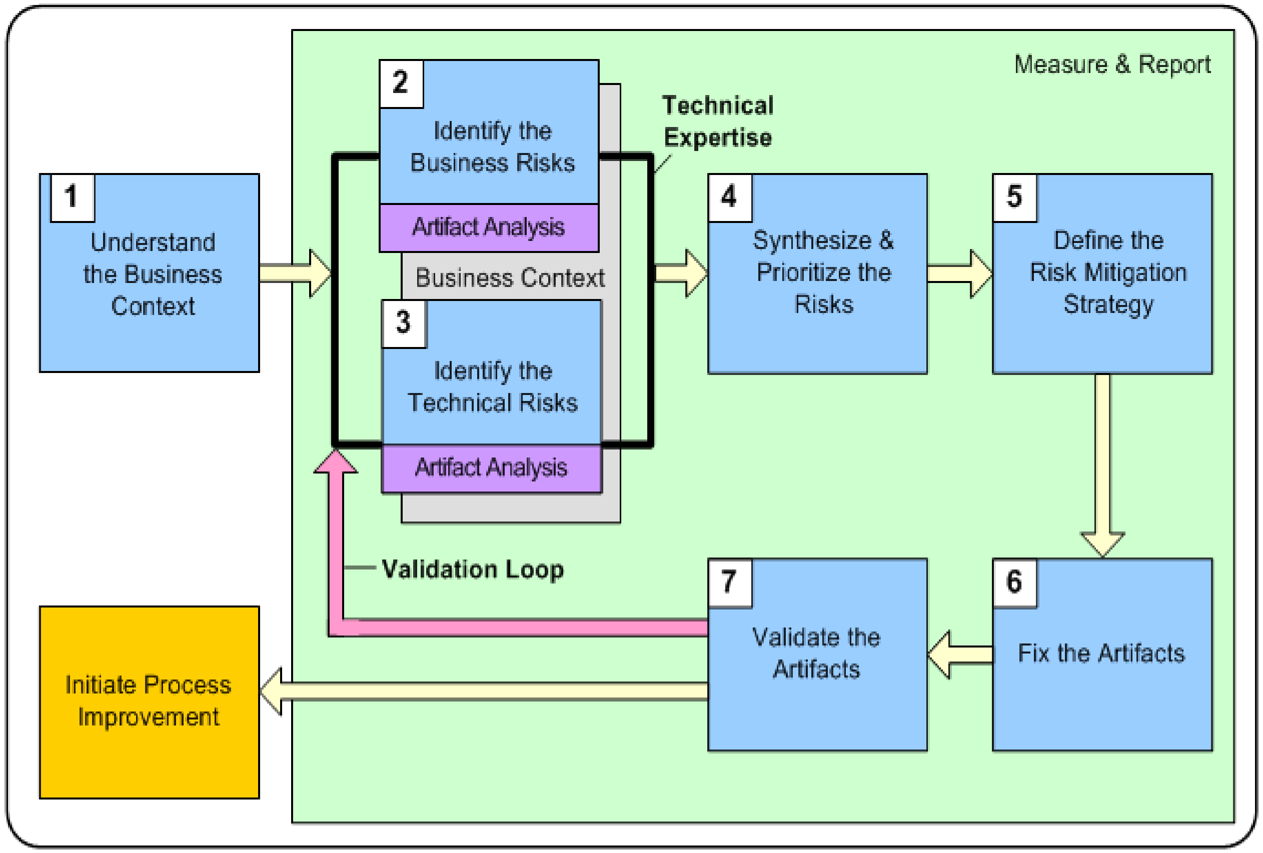
Why risk management?

- Business understands the idea of risk, even software risk
- Technical perfection is impossible
  - There is no such thing as 100% security
  - Perfect quality is a myth
- Technical problems do not always spur action
  - Answer the “So what?” question explicitly
- Help customers understand what they should *do* about software risk
- Build better software

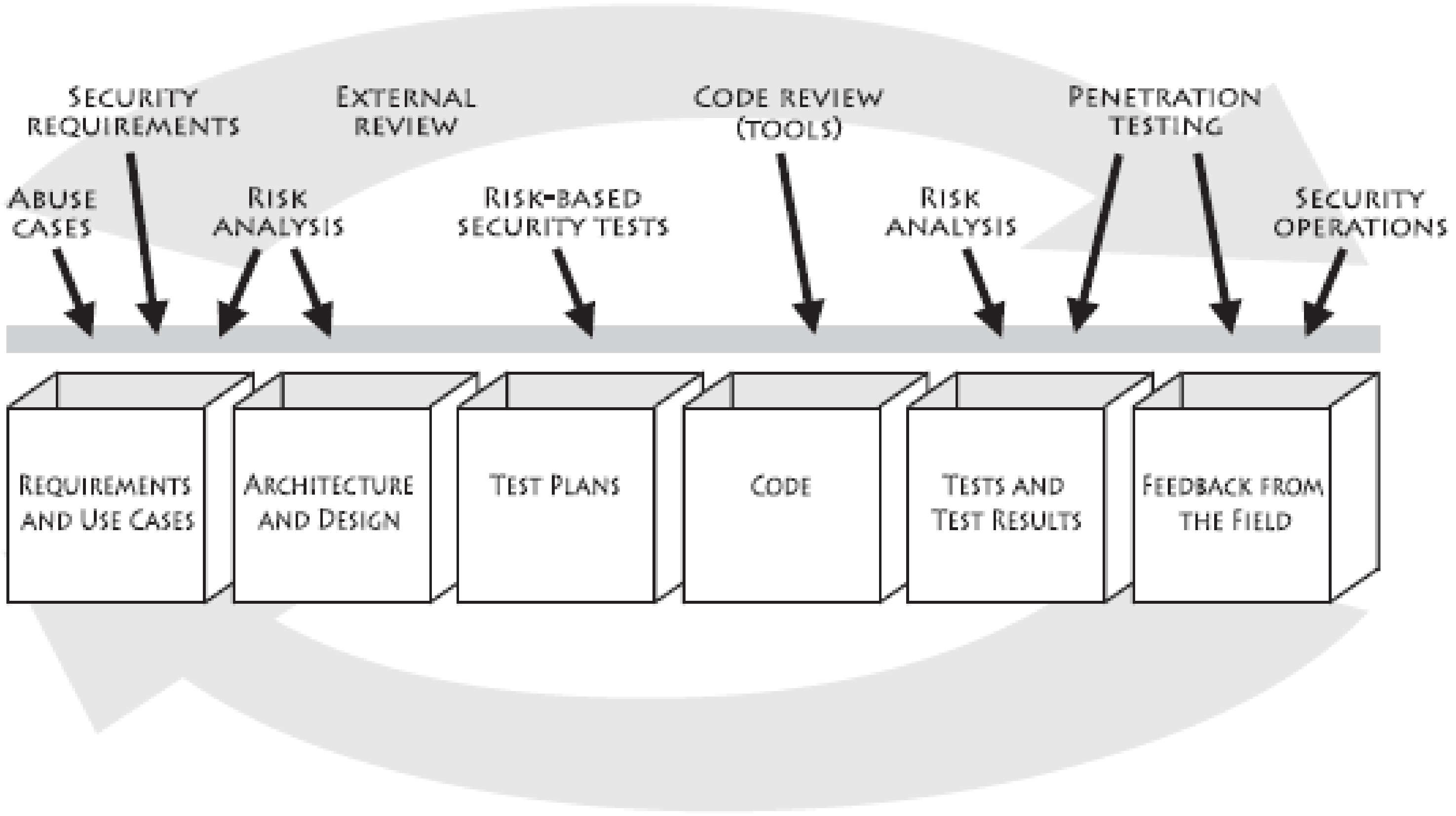
So what?

cigital KRvW Associates

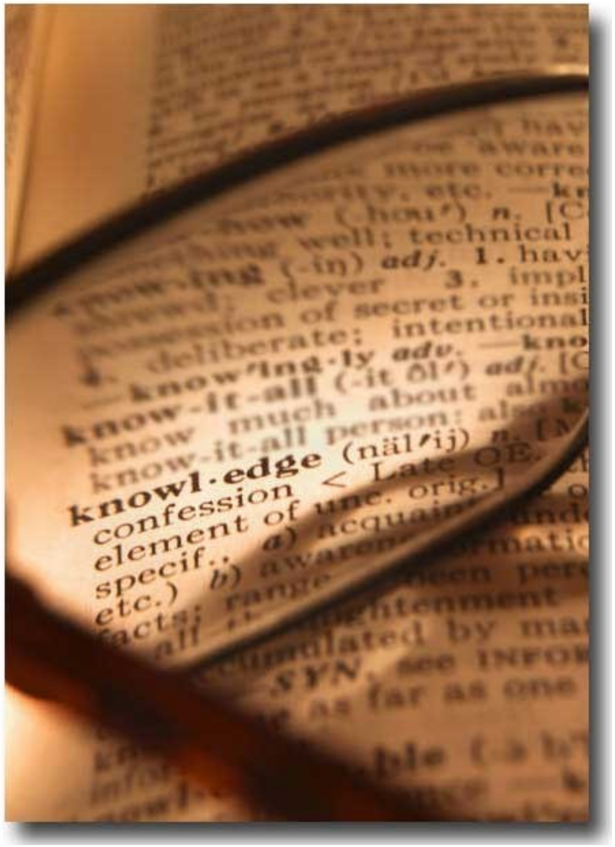
The Cigital risk management framework


Measure & Report
Technical Expertise
1 Understand the Business Context
2 Identify the Business Risks
Artifact Analysis
Business Context
3 Identify the Technical Risks
Artifact Analysis
4 Synthesize & Prioritize the Risks
5 Define the Risk Mitigation Strategy
6 Fix the Artifacts
7 Validate the Artifacts
Validation Loop
Initiate Process Improvement

Software security touchpoints



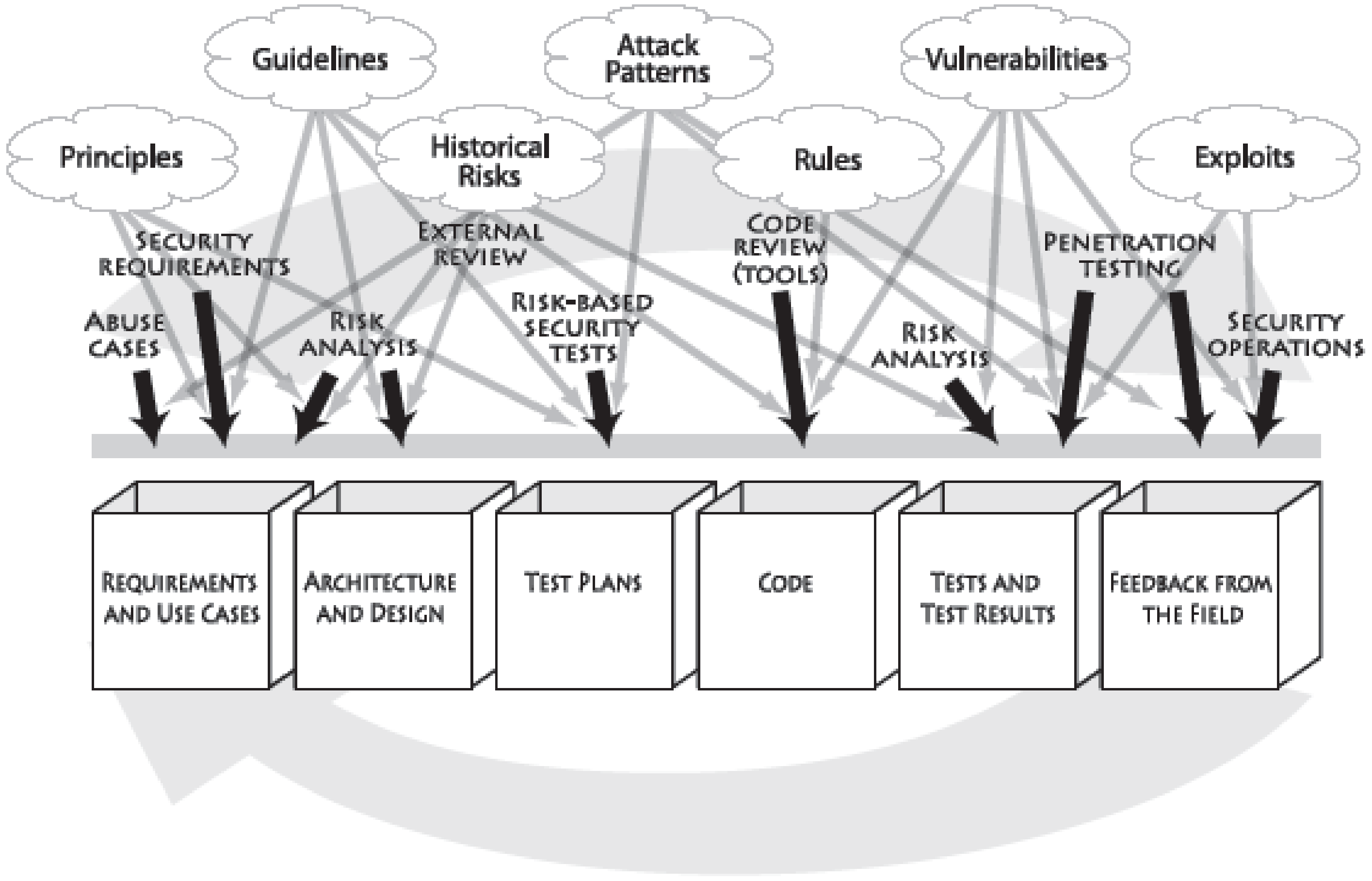
Knowledge catalogs



- Principles
- Guidelines
- Rules
- Attack patterns
- Vulnerabilities
- Historical Risks

KRvW Associates

Knowledge map



cigital KRvW Associates

Managing knowledge

- Perhaps the toughest hurdle
  - Combines people, skills, experience, etc.
  - Training helps, but there is no substitute for experience
- Start with clear targets in mind
  - Train to get started
  - Hire qualified people
- Mentoring is vital
  - Apprenticeship still plays its roll

Will this stuff work?

When applied thoughtfully, there is no reason that you can't produce measurable improvements in your software

- Don't get too hung up on process
- Take small steps towards your goal
- Start measuring immediately

If you can't measure it, how can you manage it?