A Software Security Risk Classification System

Eric Dalci
Sr. Security Consultant
edalci@cigital.com

Robert Hines
Managing Consultant
rhines@cigital.com

cigital

Software Confidence. Achieved.

Metricon 2.0 - August 2007

www.cigital.com
info@cigital.com
+1.703.404.9293

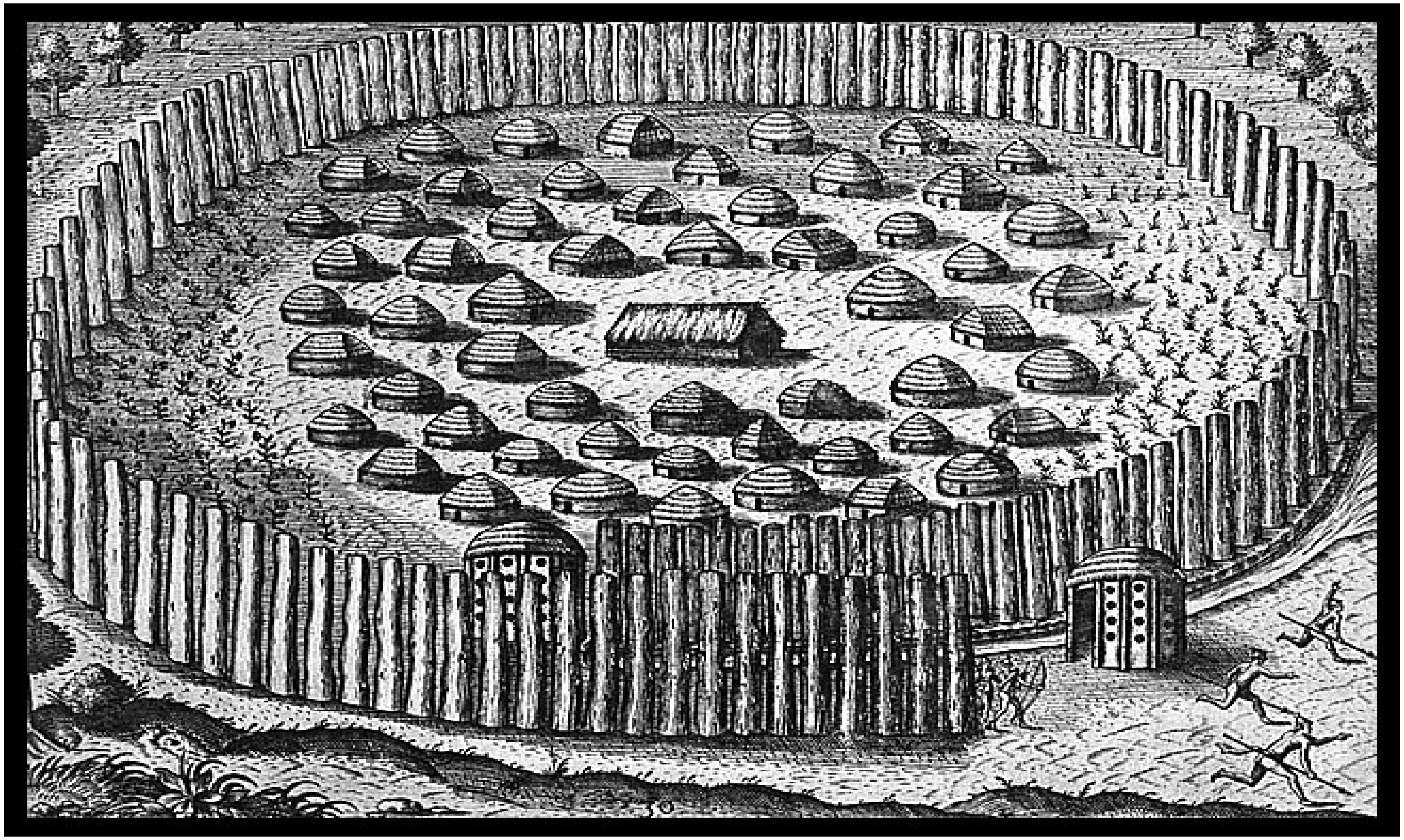
Tuesday, August 07, 2007

Agenda

- Goals and purpose of RCS
- Context
- Risk Evaluation
  - Security Metrics/Factors
- Risk Classification
- Preliminary Results
- Conclusion

cigital

First, an analogy…



The Indian village needs to fortify its huts. Where to start first ?

Image Source : http://pelotes.jea.com/NativeAmerican/LeMoyne/FORTIFIED%20TOWNS.GIF

Goals and purpose of RCS

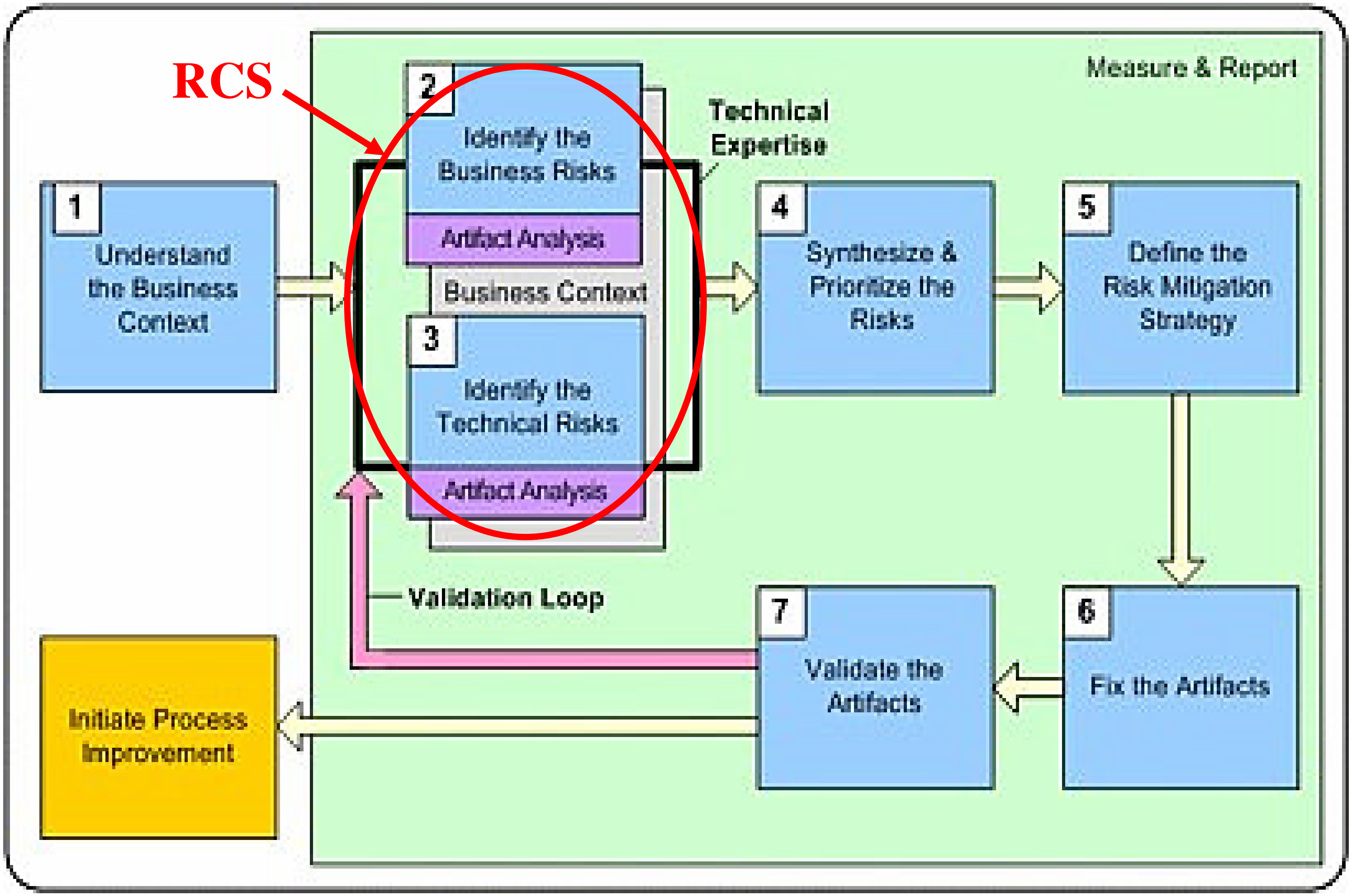
- Estimation practice of application’s potential risk (system’s insecurity) with respect to other systems in the portfolio, quickly and with nominal level of effort.
- Determination of what SLDC actions to require for systems with a given risk profile

cigital

Outcome of RCS

- Prioritization of application portfolio
  - Segregate different risk profile (High, Medium, Low)
- Portfolio Risk Evaluation
  - Identify weaknesses across portfolio
- Applicable Risk Mitigation
  - Depending on the risk profile and Lifecycle stage apply set of mitigation practices.

Cigital Risk Management Framework


RCS
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

Category of Risk

Categories of Risk	**Description**
Business Risk – Risks Inflicted upon the System by External Parties	
Market/User	Issues with the desires, requirements and satisfaction of the end users of the system
Resource (availability & capability)	Issues with Staff, Capabilities, Budget, etc.
Technical Risk – Risks Experienced as a Result of Direct System Activities	
Architecture & Design	Issues with the system architecture and design
Implementation	Issues with the technology stack used to implement the system
Quality	Issues with the accuracy, reliability and predictability of the system
Security	Issues with the confidentiality, integrity and availability of the system and its data
Operations & Maintenance	Issues with the operation and maintenance of the deployed system

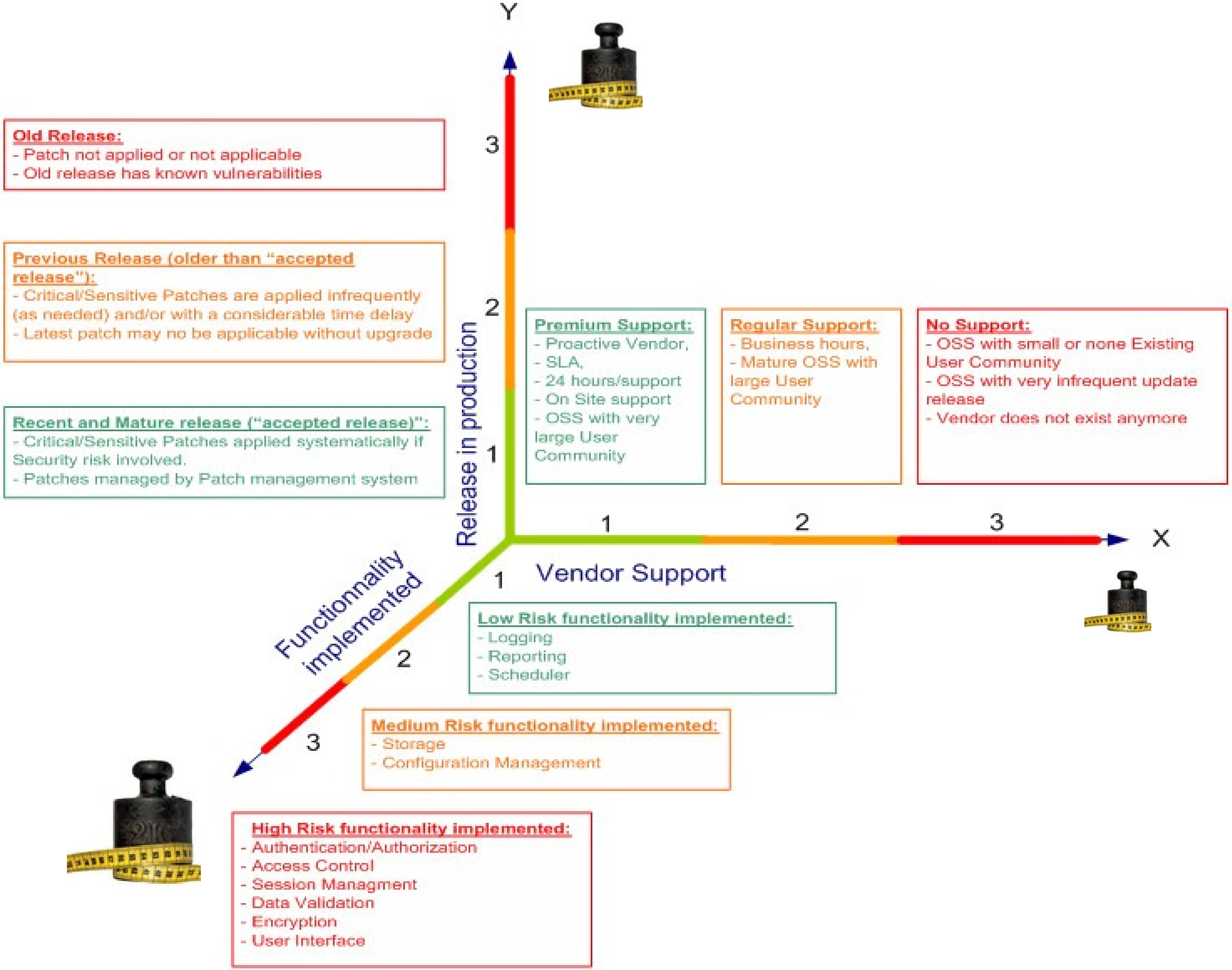
Factors

- **Business Risk**
  - Corollary impacts
  - Data Sensitivity
  - Sunk Level of Effort
  - Production Failure
  - User Count
  - User Domain
- **Technical Risk**
  - Third party COTS/OSS
  - Code Size
  - Defect Density
  - Web Vulnerability Results
  - Static Analysis Tools Results
  - Competency in Technology

Data Sensitivity	Score
Public	1
Internal Use Only	2
Confidential	3
Confidential restricted	4

Number of Users	Score
n/a	0
< 50 – Department	1
< 500 – Business unit	2
< 10,000 – Company wide	3
> 10,000 – General public	4

Measuring the COTS/OSS Factor



Old Release:
- Patch not applied or not applicable
- Old release has known vulnerabilities

Previous Release (older than "accepted release"):
- Critical/Sensitive Patches are applied infrequently (as needed) and/or with a considerable time delay
- Latest patch may no be applicable without upgrade

Recent and Mature release ("accepted release"):
- Critical/Sensitive Patches applied systematically if Security risk involved.
- Patches managed by Patch management system

Premium Support:
- Proactive Vendor,
- SLA,
- 24 hours/support
- On Site support
- OSS with very large User Community

Regular Support:
- Business hours,
- Mature OSS with large User Community

No Support:
- OSS with small or none Existing User Community
- OSS with very infrequent update release
- Vendor does not exist anymore

Low Risk functionality implemented:
- Logging
- Reporting
- Scheduler

Medium Risk functionality implemented:
- Storage
- Configuration Management

High Risk functionality implemented:
- Authentication/Authorization
- Access Control
- Session Managment
- Data Validation
- Encryption
- User Interface

Factors that we dropped

- Cyclomatic complexity
  - Code basis heterogeneous (.NET, Java, C, etc.)
- Process related metrics
  - Organization is not using consistent security processes across projects.
- Other Factors which would return subjective answers or expensive to collect.
- Poor results with “Competency in Technology”

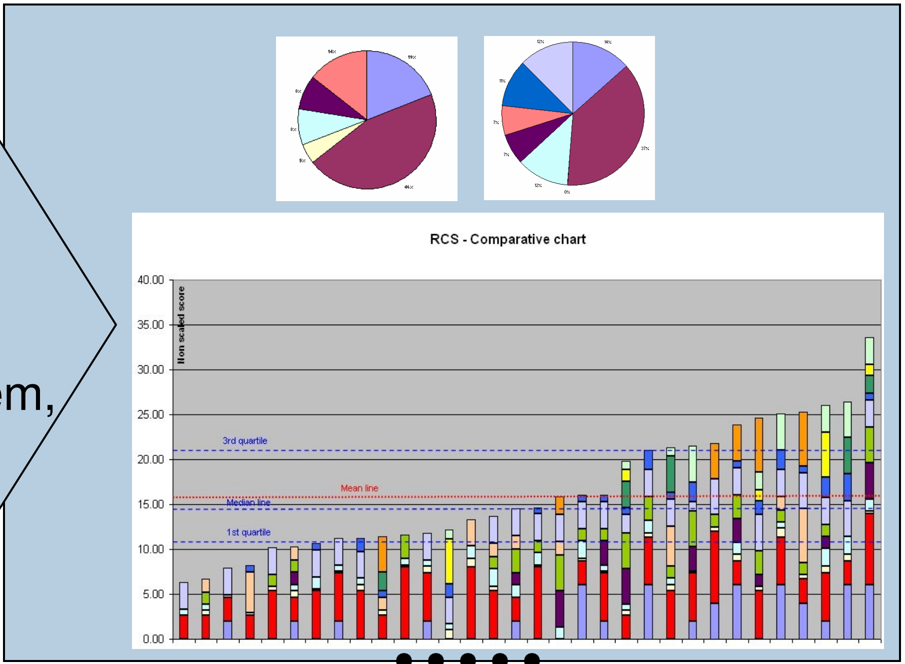
Portfolio ranking

Analysis

- Portfolio Risk Distribution
- Standard Deviation
- Correlation Matrix

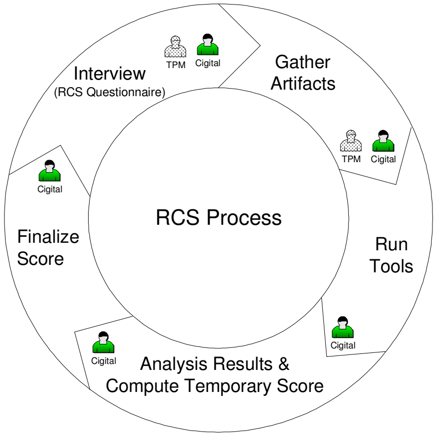
System Inputs

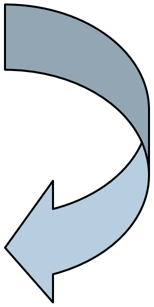
- Questionnaire,
- Tools,
- Defect tracking system,
- etc.



Calibration

- Weights
- Scale
- Pairwise Comparison





Feedback loop

Portfolio segregation

- Which Systems had **high score** ?
  - Web facing Systems
  - Large code size applications
  - Complex applications
  - New applications (No DR, new Technology, etc.)

- Which Systems had **low score** ?
  - Low user count and/or Internal applications
  - Low corollary impacts (downstream impacts)
  - Small code size applications

cigital

Calibration (Weight Systems)

Measure	**Weight**	**Correlation with aggregated score**
Corollary Impacts	1.5	0.39
Data Sensitivity	2	0.07
Sunk Level of Effort	0.25	0.35
Production Failure	0.5	0.11
User Domain	1	0.36
User Count	1	**0.49**
Total Business Risk	6.25	**0.58**
Competency in technology	1.5	0.19
Third party COTS/OSS	1	0.29
Code Size	0.75	**0.60**
Defect Density	1	0.27
Web Vulnerability Results	1.25	0.28
Static Analysis Tool Results	1	**0.60**
Contingency plan	1.5	0.41
Total Technical Risk	8	**0.73**

Conclusion

- Heuristic approach
- Preliminary results reflect expert’s opinion
- Calibration specific to your organization

- Questions ?