JOURNAL OF OBJECT TECHNOLOGY

Online at www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2003**

Vol. 2, No. 1, January-February 2003

Engineering Security Requirements

Donald G. Firesmith, Firesmith Consulting, U.S.A.

Abstract

Most requirements engineers are poorly trained to elicit, analyze, and specify security requirements, often confusing them with the architectural security mechanisms that are traditionally used to fulfill them. They thus end up specifying architecture and design constraints rather than true security requirements. This article defines the different types of security requirements and provides associated examples and guildlines with the intent of enabling requirements engineers to adequately specify security requirements without unnecessarily constraining the security and architecture teams from using the most appropriate security mechanisms for the job.

1 SECURITY REQUIREMENTS

The engineering of the requirements for a business, system or software application, component, or (contact, data, or reuse) center involves far more than merely engineering its functional requirements. One must also engineer its quality, data, and interface requirements as well as its architectural, design, implementation, and testing constraints. Whereas some requirements engineers might remember to elicit, analyze, specify, and manage such quality requirements as interoperability, operational availability, performance, portability, reliability, and usability, many are at a loss when it comes to security requirements. Most requirements engineers are not trained at all in security, and the few that have been trained have only been given an overview of security architectural mechanisms such as passwords and encryption rather than in actual security requirements. Thus, the most common problem with security requirements, when they are specified at all, is that they tend to be accidentally replaced with security-specific architectural constraints that may unnecessarily constrain the security team from using the most appropriate security mechanisms for meeting the true underlying security requirements. This article will help you distinquish between security requirements and the mechanisms for achieving them, and will provide you with good examples of each type of security requirement.

In today's world of daily virus alerts, malicious crackers, and the threats of cyber-terrorism, it would be well to remember the following objectives of security requirements:

Cite this column as follows: Donald Firesmith: *Engineering Security Requirements*, in Journal of Object Technology, vol. 2, no. 1, January-February 2003, pages 53-68.
http://www.jot.fm/issues/issue_2003_01/column6

- Ensure that users and client applications are identified and that their identities are properly verified.
- Ensure that users and client applications can only access data and services for which they have been properly authorized.
- Detect attempted intrusions by unauthorized persons and client applications.
- Ensure that unauthorized malicious programs (e.g., viruses) do not infect the application or component.
- Ensure that communications and data are not intentionally corrupted.
- Ensure that parties to interactions with the application or component cannot later repudiate those interactions.
- Ensure that confidential communications and data are kept private.
- Enable security personnel to audit the status and usage of the security mechanisms.
- Ensure that applications and centers survive attack, possibly in degraded mode.
- Ensure that centers and their components and personnel are protected against destruction, damage, theft, or surreptitious replacement (e.g., due to vandalism, sabotage, or terrorism).
- Ensure that system maintenance does not unintentionally disrupt the security mechanisms of the application, component, or center.

To meet the above objectives, we will briefly address each of the following corresponding kinds of security requirements:

- Identification Requirements
- Authentication Requirements
- Authorization Requirements
- Immunity Requirements
- Integrity Requirements
- Intrusion Detection Requirements
- Nonrepudiation Requirements
- Privacy Requirements
- Security Auditing Requirements
- Survivability Requirements
- Physical Protection Requirements
- System Maintenance Security Requirements

Guidelines

The following guidelines have proven useful with eliciting, analyzing, specifying, and maintaining security requirements:

- **Security Policy**
  A security requirement is typically a detailed requirement that implements an overriding security policy.

- **Correctness Requirements**
  Security requirements depend on correctness requirements because implementation defects are often bugs that produce security vulnerabilities. Thus, using a type-unsafe languages such as C can result in array boundary defects that can be exploited to run malicious scripts.
- **Feasibility**
  It is impossible to build a 100% secure business, application, component, or center. Increasing security typically:
  - Increases the associated cost.
  - Increases the associated schedule.
  - Decreases the associated usability.
- **Misuse Cases**
  Whereas functional requirements are now typically specified as use cases, traditional narrative English language security requirements can often be analyzed, refined, and thus further specified as misuse or abuse cases [Sindre and Opdahl 2001] [Alexander2003] whereby:
  - The **user** client external (human actor or application) of a use case is replaced by a **misuser** (e.g., cracker or disgruntled employee) who attempts to violate the security of an application, component, or center.
  - The normal user-initiated interactions of the user case are replaced by the misuser-initiated attack interactions of the misuse case.
  - The required application or component response interactions and postconditions of the user case are replaced by the required application or component security-oriented responses and postconditions of the misuse case.
  - Note that whereas goals drive use case requirements, threats drive misuse case requirements.
  - Note also that a very common problem with using misuse cases as a requirements approach is that they often assume the prior existance of architectural security mechanisms to be thwarted, and thus misuse cases may be better suited for security threat analysis and the generation of security test cases than for specifying security requirements. If misuse cases are to be used for requirements, they should be 'essential' misuse cases that do not contain unnecessary architecture and design constraints.
- **Threats vs. Goals**
  Whereas most requirements are based on higher level goals, security requirements are driven by security threats. Thus, whereas most requirements are stated in terms of what must happen, security requirements are often specified in terms of what must not be allowed to happen. Part of security engineering is therefore similar to (and can be thought of as a specialized form of) risk management. Therefore, base the security requirements on the results of a thorough security risk assessment by the security team. Such an assessment identifies the significant threats and their associated estimated frequencies, individual losses, and yearly losses. This allows the requirements team to ensure that the security requirements are cost effective.

- **Requirements vs. Architectural Mechanisms and Design Decisions**
  Care should be taken to avoid unnecessarily and prematurely specifying architectural mechanisms for fulfilling unspecified security requirements (e.g., specifying the use of user identifiers and passwords as identification and authentication requirements). The requirements team is often not qualified to make architecture decisions, and doing so may cause problems in the relationship between the requirements team and the architecture team. Specifying security constraints may also unnecessarily prevent the architecture team from choosing different, and potentially better, security mechanisms (e.g., biometric devices such as retina scanners, fingerprint readers) to meet the real underlying security requirements. If specific security architectural mechanisms and designs must be specified (e.g., for legal, constractural, or similar reasons), then specify them as architectural and design constraints, not as security requirements.
- **Validating Security Requirements**
  Security requirements typically require security-specific testing in addition to the traditional types of testing. Test cases may be based on misuse cases that are analogous to the test cases developed for use case based functional testing. Also, load and stress testing can be useful for testing Denial of Service (DoS) attacks.

2 IDENTIFICATION REQUIREMENTS

An identification requirement is any security requirement that specifies the extent to which a business, application, component, or center shall identify its externals (e.g., human actors and external applications) before interacting with them.

Examples

- "The application shall identify all of its client applications before allowing them to use its capabilities."
- "The application shall identify all of its human users before allowing them to use its capabilities."
- "The data center shall identify all personnel before allowing them to enter."
- Single Sign-on) "The application shall not require an individual user to identify himself or herself multiple times during a single session."
- "The application shall ensure that the name of the employee in the official human resource and payroll databases exactly matches the name printed on the employee's social security card."
  Rationale: This is an official requirement of the United States Social Security Administration.

Guidelines

- Identification requirements are typically insufficient by themselves. They are typically necessary prerequisites for authentication requirements.
- Identification requirements can be quantified by specifying the minimum percentage of the time that identification of a specified external [type] in a specified situation shall occur.
- Identification requirements should **NOT** be specified in terms of the types of security architecture mechanisms that are typically used to implement them, e.g., not:
  - **Who You Say You Are:**
    - Name, user identifier, or national identifier (e.g., social security number).
  - **What You Have:**
    - Digital possessions such as a digital certificate or token.
    - Physical possessions such as an employee ID card, a hardware key, or a smart card enabled with a public key infrastructure (PKI).
  - **Who You Are:**
    - Physiological traits (e.g., finger print, hand print, face recognition, iris recognition, and retina scan).
    - Behavioral characteristics (e.g., voice pattern, signature style, and keystroke dynamics).
- Do **not** analyze and specify identification requirments with use cases. A very common requirements mistake is to specify the use of user identifiers and associated passwords with design-level logon use cases.
- Identification requirements must be consistent with privacy requirements, which may require the anonymity of users.

3 AUTHENTICATION REQUIREMENTS

An authentication requirement is any security requirement that specifies the extent to which a business, application, component, or center shall verify the identity of its externals (e.g., human actors and external applications) before interacting with them.

Thus, the typical objectives of a authentication requirement are to ensure that externals are actually who or what they claim to be and thereby to avoid compromising security to an impostor.

Examples

- "The application shall verify the identity of all of its users before allowing them to use its capabilities."
- "The application shall verify the identity of all of its users before allowing them to update their user information."

- "The application shall verify the identity of its user before accepting a credit card payment from that user."
- "The application shall verify the identity of all of its client applications before allowing them to use its capabilities."
- "The data center shall verify the identity of all personnel before premitting them to enter."

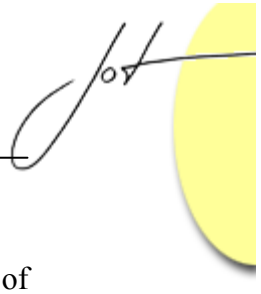
Guidelines

- Authentication depends on identification. If identity is important enough to specify, then so is authentication.
- Authentication requirements are typically insufficient by themselves, but they are necessary prerequisites for authorization requirements.
- Authentication requirements should **not** be specified in terms of the types of security architecture mechanisms that are typically used to implement them. Note that most authentication security architecture mechanisms can be used to simultaneously implement both identification and authentication requirements.
  - **Who You Know:**
    - Last four digits of your social security number, your mother's maiden name, the name of your pet, etc.
  - **What You Have:**
    - Digital possessions such as a digital certificate or token.
    - Physical possessions such as an employee ID card, a hardware key, or a smart card enabled with a public key infrastructure (PKI).
  - **Who You Are:**
    - Physiological traits (e.g., finger print, hand print, face recognition, iris recognition, and retina scan).
    - Behavioral characteristics (e.g., voice pattern, signature style, and keystroke dynamics).
- Note that some of the above authentication security architecture mechanisms can be used to simultaneously implement both identification and authentication requirements.
- Do **not** analyze and specify authentication requirments with use cases. A very common requirements mistake is to specify the use of user identifiers and associated passwords with design-level logon use cases.
- Because of the close relationship between identification and authentication requirements, they are sometimes grouped together in requirements specifications.

4 AUTHORIZATION REQUIREMENTS

An authorization requirement is any security requirement that specifies the access and usage privileges of authenticated users and client applications.

The typical objectives of an authorization requirement are to:



- Ensure that one or more persons (who have been properly appointed on behalf of the organization that owns and controls the application or component) are able to authorize specific authenticated users and client applications to access specific application or component capabilities or information.
- Ensure that specific authenticated externals can access specific application or component capabilities or information if and only if they have been explicitly authorized to do so by a properly appointed person(s).
- Thereby prevent unauthorized users from:
  - Obtaining access to inappropriate or confidential data**.**
  - Requesting the performance of inappropriate or restricted services.

Examples

- "The application shall allow each customer to obtain access to all of his or her own personal account information."
- "The application shall **not** allow any customer to access any account information of any other customer."
- "The application shall **not** allow customer service agents to access the credit card information of customers."
- "The application shall allow customer service agents to automatically email a new customer password to that customer's email address." (Note that this authorization requirement is questionable because it contains an implied authentication constraint – the use of passwords as opposed other authentication mechanisms such as digital signatures).
- "The application shall **not** allow customer service agents to access either the original or new customer password when emailing the new customer password to the customer's email address."
- "The application shall not allow one or more users to successfully use a denial of service (DoS) attack to flood it with legitimate requests of service."

Guidelines

- Authorization depends on both identification and authentication.
- Authorization requirements should **not** be specified in terms of the types of security architecture mechanisms that are typically used to implement them:
  - Authorization lists or databases.
  - Person vs. role-based vs. group-based authorization.
  - Commercial intrusion prevention systems.
  - Hardware electronic keys.
  - Physical access controls (e.g., locks, security guards).
- Authorization can be granted to:
  - Individual persons or applications.
  - Groups of related persons or applications.

- Authorization should be granted on the basis of user analysis and the associated operational requirements.
- Only a limited number of people (or roles) should be appointed to grant or change authorizations.
- A common threat to the security of an application is a denial of service (DoS) attack in which an application is flooded with legitimate requests for service. Whereas functional, operational availability, and reliability requirements cover ordinary requests for service, an additional authorization requirement may be useful because no one is authorized to flood an application with legitimate requests. Note that stress and load testing are useful for validating anti-DoS authorization requirements.

5 IMMUNITY REQUIREMENTS

An immunity requirement is any security requirement that specifies the extent to which an application or component shall protect itself from infection by unauthorized undesirable programs (e.g., computer viruses, worms, and Trojan horses).

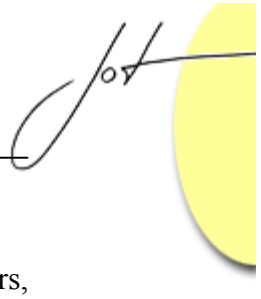
The typical objectives of an immunity requirement are to prevent any undesirable programs from destroying or damaging data and applications.

Examples

- "The application shall protect itself from infection by scanning all entered or downloaded data and software for known computer viruses, worms, Trojan horses, and other similar harmful programs."
- "The application shall disinfect any file found to contain a harmful program if disinfection is possible."
- "The application shall notify the security administrator and the associated user (if any) if it detects a harmful program during a scan."
- "To protect itself from infection by new infectious programs as they are identified and published, the application shall daily update its definitions of known computer viruses, worms, Trojan horses, and other similar harmful programs."

Guidelines

- Immunity requirements should **not** be specified in terms of the types of security architecture mechanisms that are typically used to implement them:
  - Commercial antivirus programs.
  - Firewalls.
  - Prohibition of type-unsafe languages (e.g., C) that may allow buffer overflows that contain malicious scripts.
  - Programming standards (e.g., for ensuring type safety and array bounds checking).



- Applications can delegate immunity requirements to their containing data centers, but only if those data centers provide (and will continue to provide) adequate security mechanisms to fulfill the requirements. This would be a legitimate architectural decision under certain circumstances.

6 INTEGRITY REQUIREMENTS

An integrity requirement is any security requirement that specifies the extent to which an application or component shall ensure that its data and communications are not intentionally corrupted via unauthorized creation, modification, or deletion.

The typical objectives of an integrity requirement are to ensure that communications and data can be trusted.

Examples

- "The application shall prevent the unauthorized corruption of emails (and their attachments, if any) that it sends to customers and other external users."
- "The application shall prevent the unauthorized corruption of data collected from customers and other external users."
- "The application shall prevent the unauthorized corruption of all communications passing through networks that are external to any protected data centers."

Guidelines

- Integrity requirements should **not** be specified in terms of the types of security architecture mechanisms that are typically used to implement them:
  - Cryptography.
  - Hash Codes.

7 INTRUSION DETECTION REQUIREMENTS

An intrusion detection requirement is any security requirement that specifies the extent to which an application or component shall detect and record attempted access or modification by unauthorized individuals.

The typical objectives of an intrusion detection requirement are to:

- Detect unauthorized individuals and programs that are attempting to access the application or component.
- Record information about the unauthorized access attempts.
- Notify security personnel so that they can properly handle them.

Examples

- "The application shall detect and record all attempted accesses that fail identification, authentication, or authorization requirements."
- "The application shall daily notify the data center security officer of all failed attempted accesses during the previous 24 hours."
- "The application shall notify the data center security officer within 5 minutes of any repeated failed attempt to access the employee and corporate financials databases."

Guidelines

- Intrusion detection requirements depend on identification, authentication, and authorization requirements.
- Intrusion detection requirements should **not** be specified in terms of the types of security architecture mechanisms that are typically used to implement them:
  - Alarms.
  - Event Reporting.
  - Use of a specific commercial-off-the-shelf (COTS):
    - Intrusion Detection System (IDS).
    - Intrusion Prevention System (IPS).

8 NONREPUDIATION REQUIREMENTS

A nonrepudiation requirement is any security requirement that specifies the extent to which a business, application, or component shall prevent a party to one of its interactions (e.g., message, transaction) from denying having participated in all or part of the interaction.

The typical objectives of a nonrepudiation requirement are to:

- Ensure that adequate tamper-proof records are kept to prevent parties to interactions from denying that they have taken place.
- Minimize any potential future legal and liability problems that might result from someone disputing one of their interactions.

Examples

- "The application shall make and store tamper-proof records of the following information about each order received from a customer and each invoice sent to a customer:
  - The contents of the order or invoice.
  - The date and time that the order or invoice was sent.
  - The date and time that the order or invoice was received.
  - The identity of the customer. "

Guidelines

- Nonrepudiation requirements primarily deal with ensuring that **adequate tamper-proof records** are kept. It is insufficient to merely make records; these records must be complete and tamperproof.
- Nonrepudiation requirements typically involve the storage of a significant amount of information about each interaction including the:
  - Authenticated identity of all parties involved in the transaction.
  - Date and time that the interaction was sent, received, and acknowledged (if relevant).
  - Significant information that is passed during the interaction.
- Nonrepudiation requirements are based on, can be specified in reference to, and should not redundantly specify:
  - Functional requirements specifying mandatory interactions.
  - Data requirements specifying the data that is stored and passed with these interactions. Note that nonrepudiation requirements may add making the data tamperproof.
- Nonrepudiation requirements are related to, but potentially more restrictive than auditability requirements.
- Nonrepudiation requirements should **NOT** be confused with (and specified in terms of) the security mechanisms that can be used to implement them:
  - Digital signatures (to identify the parties).
  - Timestamps (to capture dates and times).
  - Encryption and decryption (to protect the information).
  - Hash functions (to ensure that the information has not been changed).

9 PRIVACY REQUIREMENTS

A privacy requirement is any security requirement that specifies the extent to which a business, application, component, or center shall keep its sensitive data and communications private from unauthorized individuals and programs.

The typical objectives of a privacy requirement are to:

- Ensure that unauthorized individuals and programs do not gain access to sensitive data and communications.
- Provide access to data and communications on a "need to know" basis.
- Minimize potential bad press, loss of user confidence, and legal liabilities.

Examples

- **Anonymity.**
  - "The application shall not store any personal information about the users."

- **Communications Privacy.**
  - "The application shall not allow unauthorized individuals or programs access to any communications."
- **Data Storage Privacy.**
  - "The application shall not allow unauthorized individuals or programs access to any stored data."

Guidelines

- Privacy requirements should clearly identify their scope:
  - The specific data and communications that are sensitive, confidential, trade secrets, etc.
  - The specific places where this communication takes place (e.g., over the Internet, outside of a secure data center).
- Privacy requirements are related to, but go beyond, requirements, because people and applications should have access only to the data and communications for which they are authorized.
- Privacy requirements may overlap certain legal constraints such as laws that require certain data (e.g., credit card information) to be kept private.
- Privacy requirements should **not** be confused with (nor specified in terms of) the architectural security mechanisms that can be used to implement them:
  - Public or private key encryption and decryption.
  - Commercial-off-the-shelf cryptography packages.
- Privacy requirements must be consistent with auditability requirements, identification requirements, and nonrepudiation requirements, which require users to be identified and information about their interactions to be stored. For example, consider a privacy-oriented eMarketplace application that acts as an intermediary between buyers, merchants, and a credit card authorization processing gateway. The buyers may not want to provide private personal information (e.g., their name, billing address, credit card number and expiration date) to merchants who do not really need it if they are not going to be the ones to obtain purchase authorizations from the credit card authorization processors. Note that electronic wallets undermine privacy because they make it easy for buyers to supply private information to merchants. Instead, the eMarketplace strongly supports privacy by:
  - Hiding private customer personal information from merchants.
  - Authorizing the credit card purchase for the buyer (which is why the merchant wants the private information).
  - Only suppling the merchant with the non-private information (e.g., delivery address and credit payment information, such as credit approval).
  - Strongly encrypting all communications and storage of private information.

10 SECURITY AUDITING REQUIREMENTS

A security auditing requirement is any security requirement that specifies the extent to which a business, application, component, or center shall enable security personnel to audit the status and use of its security mechanisms.

The typical objectives of a security auditing requirement are to ensure that the application or component collects, analyzes, and reports information about the:

- Status (e.g., enabled vs. disabled, updated versions) of its security mechanisms.
- Use of its security mechanisms (e.g., access and modification by security personnel).

Examples

- "The application shall collect, organize, summarize, and regularly report the status of its security mechanisms including:
  - Identification, Authentication, and Authorization.
  - Immunity.
  - Privacy.
  - Intrusion Detection."

Guidelines

- Care should be taken to avoid unnecessary duplication between security-auditing and intrusion detection requirements.
- Security auditing requirements should **not** be confused with (nor specified in terms of) the architectural security mechanisms that can be used to implement them:
  - Audit Trails.
  - Event Logs.

11 SURVIVABILITY REQUIREMENTS

A survivability requirement is any security requirement that specifies the extent to which an application or center shall survive the intentional loss or destruction of a component.

The typical objective of a survivability requirement is to ensure that an application or center either fails gracefully or else continues to function (possibly in a degraded mode), even though certain components have been intentionally damaged or destroyed.

Examples

- "The application shall not have a single point of failure."

- "The application shall continue to function (possibly in degraded mode) even if a data center is destroyed."

Guidelines

- Survivability requirements are often critical for military applications.
- Avoid confusing robustness requirements with survivability requirements. Survivabilty requirements deal with safeguarding against damage or loss due to *intentional* malicious threats, whereas robustness requirements deal with safeguarding against *unintentional* hardware failures, human errors, etc.
- Survivability requirements should **NOT** be confused with (nor specified in terms of) the architectural security mechanisms that can be used to implement them:
  - Hardware redundancy.
  - Data center redundancy.
  - Failover software.

12 PHYSICAL PROTECTION REQUIREMENTS

A physical protection requirement is any security requirement that specifies the extent to which an application or center shall protect itself from physical assault.

The typical objectives of physical protection requirements are to ensure that an application or center are protected against the physical damage, destruction, theft, or replacement of hardware, software, or personnel components due to vandalism, sabotage, or terrorism.

Examples

- "The data center shall protect its hardware components from physical damage, destruction, theft, or surreptitious replacement."
- "The data center shall protect its personnel from death, injury, and kidnapping."

Guidelines

- Physical protection requirements are related to survivability requirements. Survivability requirements specify continued functioning after an attack, whereas physical protection requirements specify the protection of components. Physical protection requirements are typically prerequisites for survivability requirements.
- Physical protection requirements should **NOT** be confused with (nor specified in terms of) the architectural security mechanisms that can be used to implement them:
  - Locked Doors.
  - Security Guards.
  - Rapid Access to Police.

13 SYSTEM MAINTENANCE SECURITY REQUIREMENTS

A system maintenance security requirement is any security requirement that specifies the extent to which an application, component, or center shall prevent *authorized* modifications (e.g., defect fixes, enhancements, updates) from accidentally defeating its security mechanisms.

The typical objective of a system maintenance security requirement is to maintain the levels of security specified in the security requirements during the usage phase.

Examples

- "The application shall not violate its security requirements as a result of the upgrading of a data, hardware, or software component."
- "The application shall not violate its security requirements as a result of the replacement of a data, hardware, or software component."

Guidelines

- System maintenance security requirements may conflict with operational availability requirements, in that the operational availability requirements may not allow one to take the application or component off-line during maintenance and the repetition of security testing.
- System maintenance security requirements should **NOT** be confused with (nor specified in terms of) the architectural security mechanisms that can be used to implement them:
  - Maintenance and enhancement procedures.
  - Associated training.
  - Security regression testing.

14 CONCLUSION

This column has addressed the need to systematically analyze and specify real security requirements as part of the quality requirements for a business, application, component, or center. It has identified and defined the different kinds of security requirements, provided good examples that may be copied, and listed guidelines that have proven useful when eliciting, analyzing, specifying, and maintaining security requirements.

In the next column, I will discuss the need to produce multiple versions of requirements specifications based on the varying needs of their intended audiences. I will also provide criteria for evaluating requirements specification and management tools based on this need.

REFERENCES

The contents of this column have been collected from the following sources:

[Alexander2003] Ian Alexander: Misuse Case Help To Elicit Nonfunctional Requirements, IEE CCEJ, 2001, http://easyweb.easynet.co.uk/~iany/consultancy/papers.htm.

[Firesmith2001] Donald Firesmith and Brian Henderson-Sellers: The OPEN Process Framework, Addison-Wesley-Longman, 2001.

[Firesmith2003a] Donald Firesmith and Didar Zowghi: Requirements Engineering: A Framework-Based Handbook, 2003.

[Firesmith2003b] Donald Firesmith: OPEN Process Framework (OPF) Website, www.donald-firesmith.com.

[Sindre and Opdahl 2001] Guttorm Sindre and Andreas Opdahl: Templates for Misuse Case Description, 2001, http://www.ifi.uib.no/conf/refsq2001/papers/p25.pdf.

ACKNOWLEDGEMENTS

The contents of this article are taken from my informational website on process engineering as well as the contents of my upcoming book on requirements engineering. I would also like to acknowledge the valuable review comments of Tom Gilb and Guttorm Sindre, which significantly improved the content of this paper.

About the author

Donald Firesmith runs Firesmith Consulting, which provides consulting and training in the development of software-intensive systems. He has worked exclusively with object technology since 1984 and has written 5 books on the subject. He is currently writing a book on requirements engineering. Most recently, he has developed a 1000+ page informational website on the OPEN Process Framework at www.donald-firesmith.com. He can be reached at donald_firesmith@hotmail.com.