# Security Risks: Management and Mitigation in the Software Life Cycle

David P. Gilliam
*Jet Propulsion Laboratory, California Institute of Technology*
david.p.gilliam@jpl.nasa.gov

## Abstract

*A formal approach to managing and mitigating security risks in the software life cycle is requisite to developing software that has a higher degree of assurance that it is free of security defects which pose risk to the computing environment and the organization. Due to its criticality, security should be integrated as a formal approach in the software life cycle. Both a software security checklist and assessment tools should be incorporated into this life cycle process and integrated with a security risk assessment and mitigation tool. The current research at JPL addresses these areas through the development of a Software Security Assessment Instrument (SSAI) and integrating it with a Defect Detection and Prevention (DDP) risk management tool.*

## 1. Introduction

Information Technology (IT) security, until recently, has been relegated to an add-on requirement and design element in software engineering. With the growing number of system security defects being discovered and as the impact of malicious code grows, addressing security in the software life cycle is an increasingly crucial activity.

There are several factors that need consideration when addressing security in the life cycle such as security requirements from the various stakeholders, the environment the software will run in, including hardware and other integrated software modules, current security mitigations, expected life time usage, and so on. These factors need consideration in addressing security in the software life cycle.

The process can be aided by the use of tools. A security template for a risk management tool can aid in the process of determining what security risks need to be addressed and mitigations and their effectiveness. Use of modeling for security requirements specification and property-based testing are showing to be effective instruments for aiding in this process.

Previous papers have discussed the use of modeling and property-based testing (PBT). [1] As the complexity of software and requirements increases, software modeling becomes more difficult due to state space explosion. The model-based verification instrument seeks to breakdown the state space explosion through the use of a developed flexible modeling framework (FMF). The property-based testing instrument then verifies the security properties specified in the requirements. Both the FMF and PBT can work together or be used independently.

In addition, other tools in the research project are a security assessment checklist, training for software engineers and developers, and a vulnerability matrix that classifies vulnerabilities and identifying security property violations. These tools in conjunction with a security risk management instrument can reduce security vulnerabilities in the life cycle. The instrument has been funded by the National Aeronautics and Space Administration (NASA) Independent Verification and Validation (IV&V) Center and is shown in Figure 1.

This paper will focus on the use of a security risk management instrument, the Defect Detection and Prevention (DDP) instrument, and the integration of the previously developed tools with this instrument.

## 2. Current State of Software Security

There are several reasons for the number of vulnerabilities and security exposures in software. Two key issues are, 1) "The attitude today is that you can write any sloppy piece of code and the compiler will run diagnostics;" and the developer relies on the complier to indicate any errors, and 2) "The constant demand for novelty means that software is always in the bleeding-edge phase, when products are inherently less reliable." To these reasons should also be added lack of skilled developers and training, and lack of resources such as code analyzers that can check the security of software and systems. As McGraw points out, "Security is like fault tolerance, a system-wide emergent property that requires much advance planning and careful design."[2]

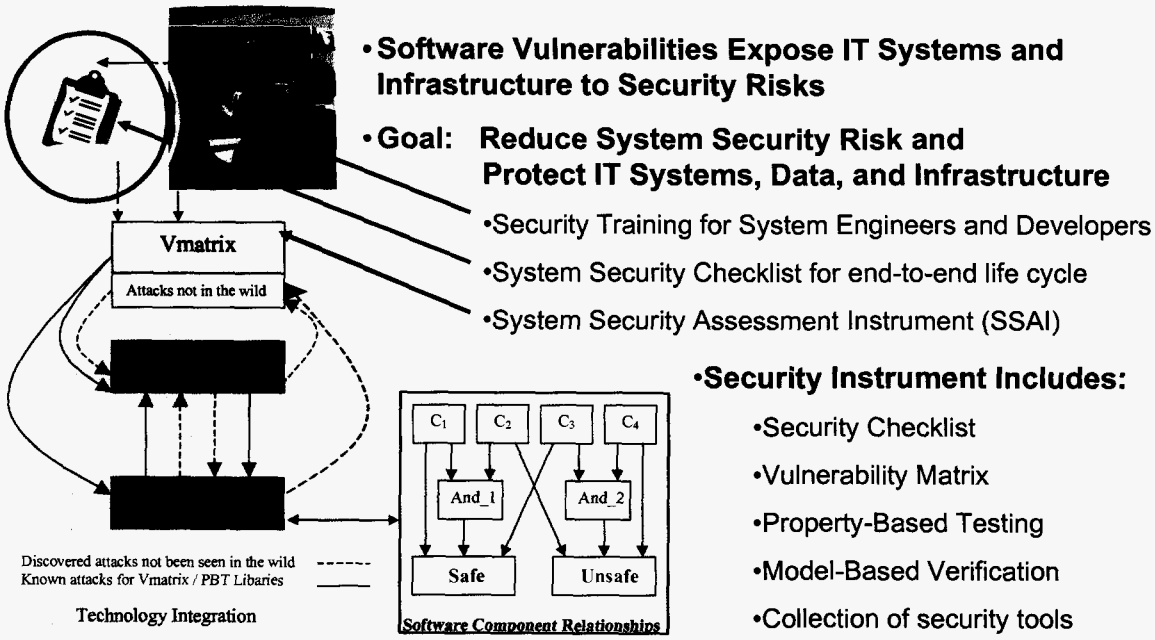


Figure 1: Software Security Assessment Instrument

## 3. Security Risk Management

The planning and design for security begins with elicitation and specification of security requirements. However, these requirements must address identified needs based on the environment and customer needs. They must also be integrated with requirements for reliability and safety. Managing and mitigating, security reliability and safety risks are three foci that require being addressed collectively in the life cycle.

The National Institute of Standards and Technology (NIST) as well as the Software Engineering Institute (SEI) address security risk in the life cycle. However, while they provide guidelines for managing security risk they fail to integrate it with other risk factors like reliability and safety. [3]

Each risk area may have an impact on the others both in risks and in mitigations. Using a risk management tool can aid in analyzing relative risks, mitigations, and their effectiveness. JPL has successfully used DDP effectively on flight projects to manage and mitigate reliability and safety risks. Not yet included is a security risk template to integrate security risk with these other risks. The DDP tool aids assessing the impact of a risk in one area on another as well as the impact of mitigations on each of these risk areas.

A reliability risk may negatively impact security such as a software system that fails insecure. Likewise a security risk may also negatively impact safety such as a weak authentication process may allow an unauthorized person to access a safety critical system such as a medical prescription database and modify data. Similarly, a security risk mitigation may positively impact safety such as a strong authentication scheme for such a system above. A security risk mitigation may negatively impact reliability as well such as anti-virus software that has to examine files for malicious content on a system that has performance parameters that may be impacted by real-time scanning. Risk assessment, analysis and mitigation is discussed in more detail in [4].

### 3.1 Assessing and Rating Security Risks – The Defect Detection and Prevention (DDP) Tool

Rating risks and potential mitigations is a difficult activity that is resource intensive. Automating as much of the process as possible is essential to managing risks. The process as shown in Figure 2 involves domain experts to evaluate the risks and a system security engineer to input the data into a database which can then be used by a risk management tool like DDP. "DDP explicitly represents risks, the objectives that risks threaten, and the mitigations available for risk reduction. By linking these three concepts, DDP is able to represent and reason about the cost-effectiveness of risk reduction alternatives." [5] "The single most important aspect of the DDP approach is that it supports multiple experts [who] pool their knowledge" allowing them "to

take the sum total of their pooled knowledge into account as they make decisions." [6]

The goal is to evaluate the relative risks and the various mitigations. DDP can discover optimal solutions to risks and mitigations in terms of effectiveness and costs by iterating through the risks and mitigations. DDP can plot data points with drill in capability to find the optimal range and what are the factors that best meet the needs of the enterprise. The tool also provides the capability to compare risks and mitigations individually (see Figure 3). DDP can then merge that data with the data for reliability and safety to analyze the change in risk factors and the impact of each domain area on the other (see Figure 4).

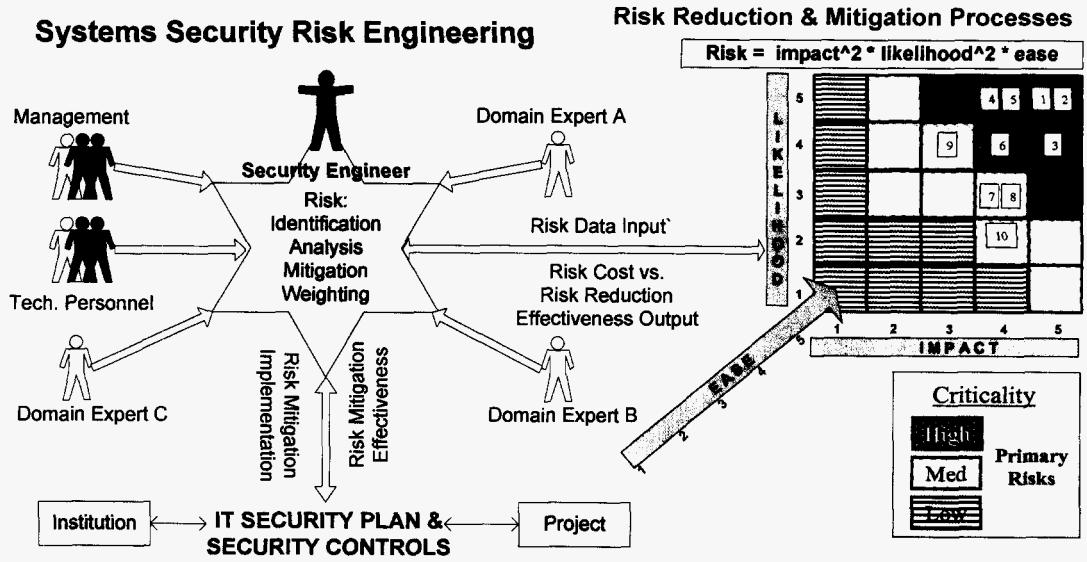


Figure. 2: Security Engineering Risk Assessment/Management Process

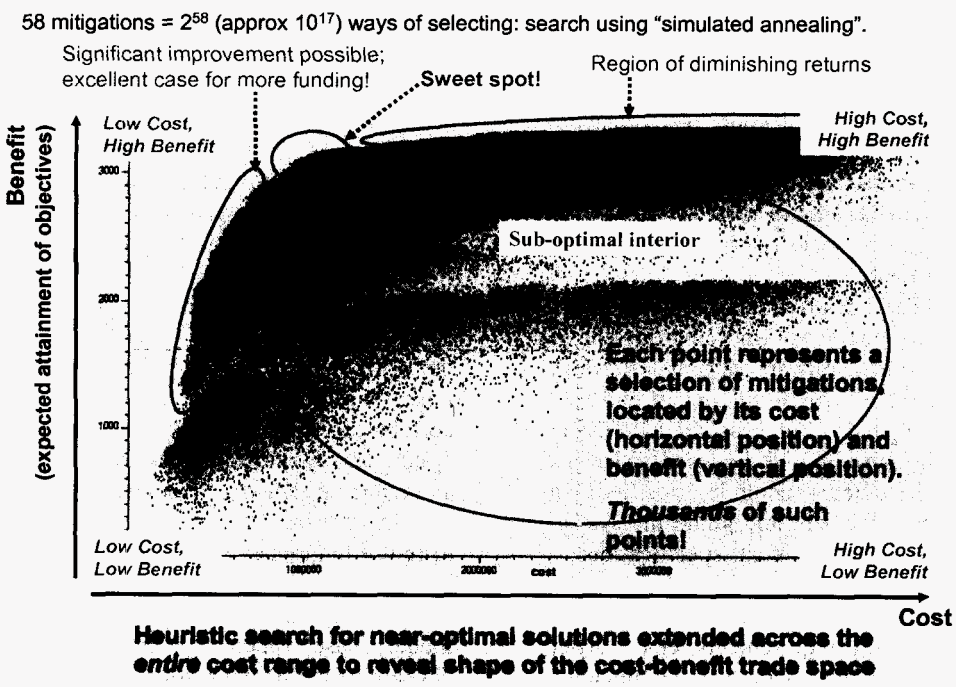


Figure. 3: Cost/Benefit trade-off analysis

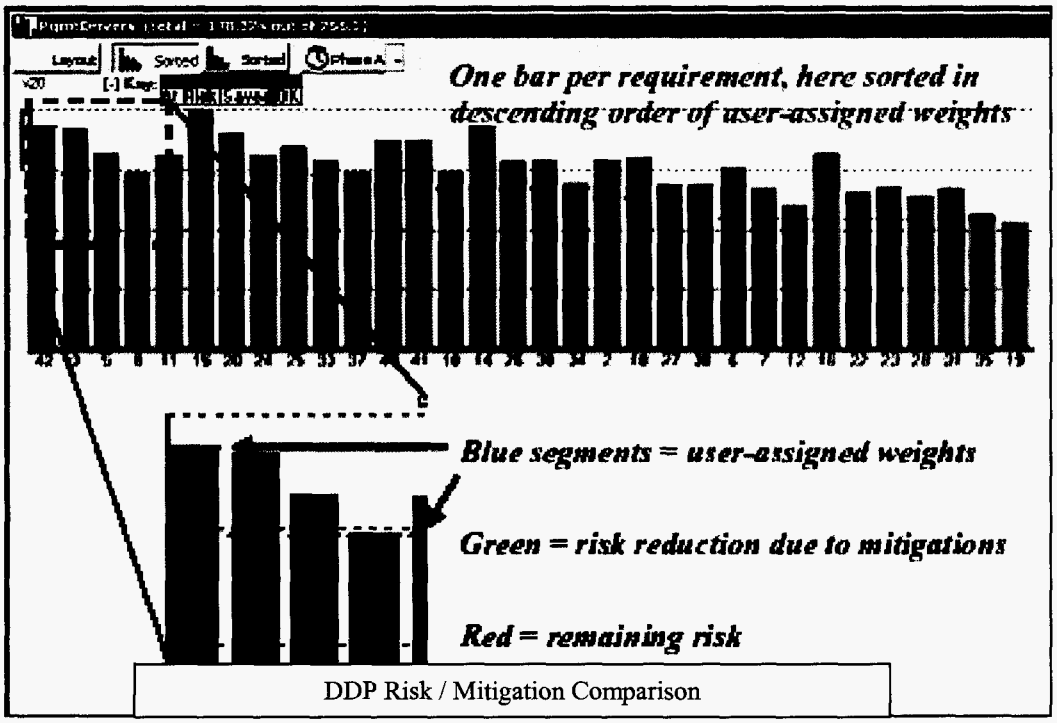


Figure.4: Graphical presentation of risks sorted into descending order of risk levels

### 3.2 Software Life Cycle Risk Mitigations

Security risk mitigations for the life cycle first require a controlled software engineering process that begins with requirements inception and follows through to decommissioning. A process for customer and stakeholder requirements specification will provide some mitigation for missed requirements and poor specifications. Mitigations can include checklists and modeling. The requirements should address institutional requirements, institutional security risks from both internal and external threats.

A security checklist can provide system engineers direction in determining the scope and environment of security settings both for software and the network systems and environment in which the system will run. The checklist should include a process for identifying stakeholders, how the system will be used, what its purpose is, etc. [7] These items should be specified in a Concept of Operations document and guide the system engineer in developing the requirements specifications.

Security requirements may include authentication, access controls to the system(s) and data, secure communications, fail secure requirements, et.al. After they have been specified, they can then be modeled using model-based verification techniques. [8] Model checking "can identify vulnerabilities and undesired exposures in software. These often arise from a number of development factors that can often be traced to poor software development practices, new modes of attacks in the network security arena, unsafe configurations, and unsafe interaction between systems and/or their components." [9]

One of the problem areas facing modeling techniques is state-space explosion where the model grows rapidly as the complexity of the system increases. [10] To address this problem, research at JPL has led to the development of the Flexible Modeling Framework (FMF). [11]

The FMF builds up a model of the system from components which are first verified for correctness using the SPIN model checker. They are then combined with other modeled components and re-verified for correctness. In this approach security properties of a system can be verified while maintaining system fidelity and reducing state-space explosion (Figure 5). [9]

An additional advantage of the FMF is in the ability to re-use the model during the maintenance phase of the life cycle where many security violations arise. Additionally, there is a potential for component and model reuse which can reduce the cost of model development on other software systems that may use these components.

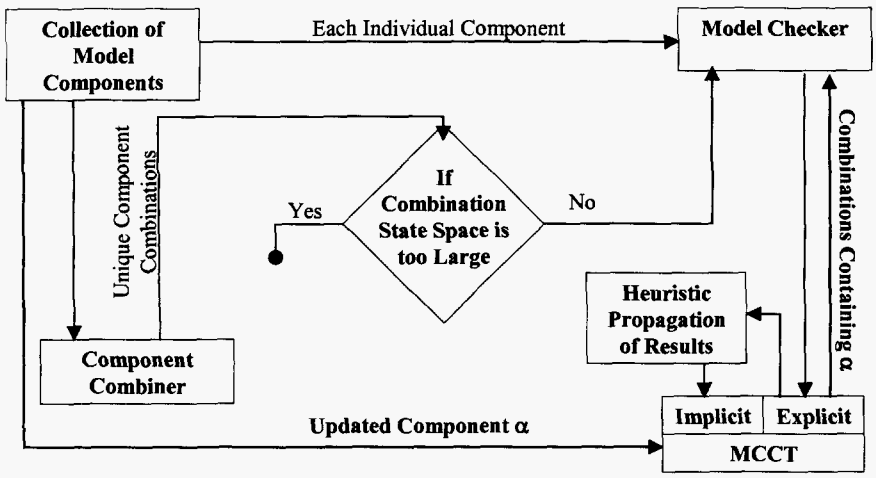


Figure.5: Illustration of the Flexible Modeling Framework process

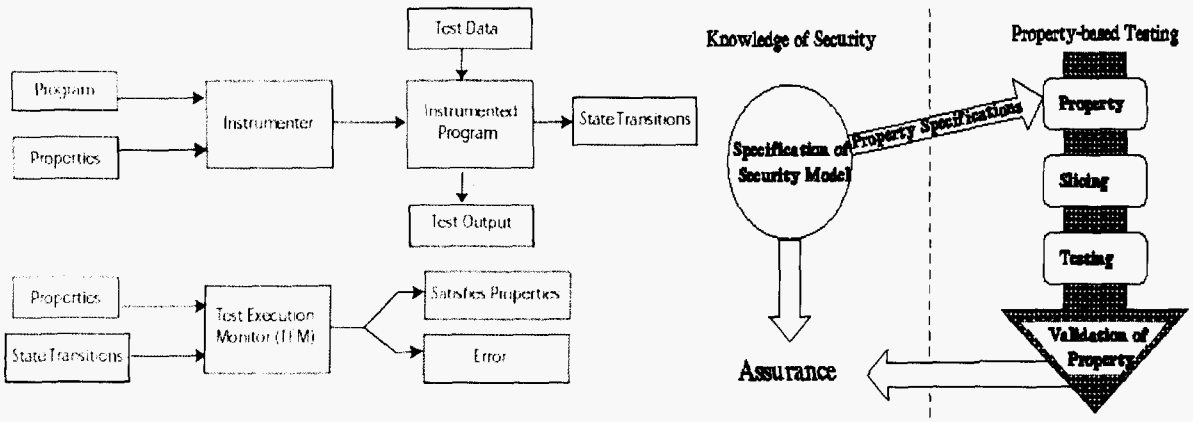


Fig. 8: PBT Process and Model

These security property specifications should then be tested during the development, test and verification, and the maintenance phases of the life cycle. Another tool being developed at JPL is a property-based testing (PBT) instrument (Figure 6). This instrument can use the security properties specified and used by the FMF. The PBT technique is code insertion as comments. The footprint adds negligible size to the code. [11] The PBT adds element to security risk mitigation by providing another element to the life cycle process. Taken together the current software security risk mitigation instrument offers a means to reduce risk in the life cycle.

Other elements that reduce risk that need to be considered in the life cycle process involve system and network environmental factors such as what other processes and services may be running on the system that may impact the system. Impacts may include the need for software to protect the system from malicious content such as anti-virus software. This software can be disk and processor resource intensive during system scanning. Impact on the software system should be considered and addressed in the risk analysis activity as part of the DDP process shown above. Further risk mitigations are the use of 'whitebox' and 'blackbox' testing, fault injection testing and vulnerability scanning of the software running in both a test environment and in its intended environment. Lastly, documentation is a large risk mitigation element. This is particularly true for use in the maintenance phase of the life cycle.

The software security assessment instrument can be used collectively or individually as needed. Figure 1, shows how the various pieces of the instrument can work in concert to provide a higher level of security assurance. Coupling the security assessment instrument

with DDP as mitigations to the life cycle, provides further benefit for being able to determine the level of risk and the effectiveness of the mitigations of the tools in a changing threat scenario.

### 3.3 Persistence of Security Risk Management

Security risks are somewhat different than other risks in that the environment is not controlled. External factors from the outside environment affect risks and their relative severity. The risks are not static. The risk threat scenario is highly volatile and is impacted by varying attack methods, including probes leading to breakins which may even be coupled with Distributed Denial of Service attacks.

The risk assessment analysis process therefore needs to be persistent and continued through the maintenance phase of the life cycle until the software and/or system are decommissioned. When decommissioning software or a system, there may exist other systems that are dependent on that software or system. Decommissioning must take into account and address these risks as well, especially if there are dependencies that rely on the software for security such as access controls.

Additionally, as the threat scenario changes risks and mitigations require re-evaluation and rating. The use of DDP can help facilitate the process by providing a process for modifying the data and re-running the process for maintenance activities to ensure that the software remains robust as new forms of attacks are discovered and exercised.

## 4. Conclusion

By addressing security risks in the software life cycle through formal approaches, the security of systems and organizations will improve greatly. By making it a persistent process, the security of software will attain a higher level of security assurance and provide the capability of responding to new emerging threats and provide a greater degree of trust in the software.

It is essential that security be addressed in the life cycle to reduce vulnerabilities and unwanted exposures in software and systems. It also gives the customers and stakeholders a greater level of assurance that the software will not pose a security risk to them and to their environment. The approach must be an inclusive formal approach that begins with requirements elicitation and specification and ends with the decommissioning of the software system.

## 5. References


[1] Gilliam, D.G., and Powell, J.D., "Integrating a Flexible Modeling Framework (FMF) with the Network Security Assessment Instrument to Reduce Software Security Risk," Proceedings of the 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), June 2002, Pittsburgh, PA.

[2] McGraw, G., "Software Risk Management for Security", Citigal White Paper, 1999, available at http://www.cigital.com/whitepapers/.

[3] Stoneburner G., Goguen, A., and Feringa, A., "Risk Management for Information Technology Systems," The National Institute of Standards and Technology Special Publication 800-30, 2001. Also, SEI, Carnegie Mellon University Software Engineering Institute, "OCTAVE Method," 11-11, 2003, available: http://www.cert.org/octave/methods.html.

[4] Gilliam, D.P., "Managing Information Technology Security Risk," International Symposium On Software Security, Tokyo Institutue of Technology, Tokyo, Japan, November 2003.

[5] Feather, M.S., Cornford, S. L., and Moran, K., "Risk-Based Analysis And Decision Making In Multi-Disciplinary Environments," Proceedings of IMECE'03 2003 ASME International Mechanical Engineering Congress & Exposition Washington, D.C., November 16–21, 2003.

[6] Cornford, S. L., Feather, M.S., Dunphy, J., Salcedo, J., and Menzies, T., "Optimizing Spacecraft Design – Optimization Engine Development: Progress and Plans," IEEE Aerospace Conference, March 2003, (available on the web at: http://ddptool.jpl.nasa.gov)

[7] Gilliam, D.G., Wolfe, T.L., Sherif., J.S., "Software Security Checklist for the Software Life Cycle," Proceedings on the 11th IEEE WETICE Workshop, June 2003, Linz, Austria., pp. 243-248

[8] Wen, W., and Mizoguchi, F., "Model checking Security Protocols: A Case Study Using SPIN, IMC," Technical Report, November, 1998

[9] Powell, J.D., Gilliam, D.P., "Component Based Model Checking," Proceedings of the 6th World Conference on Integrated Design and Process Technology, June 23-28, Pasadena CA, p66 & CD

[10] Ammann, P.E., Black, P.E., and Majurski, W., "Using Model Checking to Generate Test Specifications," 2nd International Conference on Formal Engineering Methods (1998) pp. 46-54

[11] Gilliam, D.P., Powell, J.D., Bishop, M., "Reducing Software Security Risk Through an Integrated Approach", IEEE Goddard 28th Software Engineering Workshop, Greenbelt, MD, Dec. 2003.

04-0778

**Return Fax to: +1 714 761 1784 Attn: Stephanie Kawada – WET ICE 2004**

# IEEE COPYRIGHT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

**TITLE OF PAPER/ARTICLE/REPORT/PRESENTATION/SPEECH (hereinafter, "the Work"):**

**COMPLETE LIST OF AUTHORS:**

**IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book): WET ICE 2004**

## Copyright Transfer

The undersigned hereby assigns to the Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to the above Work, and any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work. The undersigned hereby warrants that the Work is original and that he/she is the author of the Work; to the extent the Work incorporates text passages, figures, data or other material from the works of others, the undersigned has obtained any necessary permissions. **See reverse side for Retained Rights and other Terms and Conditions.**

## Author Responsibilities

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements of IEEE Policy 6.3, including provisions covering originality, authorship, author responsibilities and author misconduct. The full policy may be viewed at **http://www.ieee.org/about/whatis/policies/p6-3.html**. Authors are advised especially of IEEE Policy 6.3.1A(l): "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it."

## General Terms

- The undersigned represents that he/she has the power and authority to make and execute this assignment.
- The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
- In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall become null and void and all materials embodying the Work submitted to the IEEE will be destroyed.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.

(1)_____ _____

**Author/Authorized Agent for Joint Authors** **Date**

## U.S. Government Employee Certification (where applicable)

This will certify that all authors of the Work are U.S. government employees and prepared the Work on a subject within the scope of their official duties. As such, the Work is not subject to U.S. copyright protection.

(2)_____ _____

**Authorized Signature** **Date**

(Authors who are U.S. government employees should also sign signature line (1) above to enable the IEEE to claim and protect its copyright in international jurisdictions.)

## Crown Copyright Certification (where applicable)

This will certify that all authors of the Work are employees of the British or British Commonwealth Government and prepared the Work in connection with their official duties. As such, the Work is subject to Crown Copyright and is not assigned to the IEEE as set forth in the first sentence of the Copyright Transfer Section above. The undersigned acknowledges, however, that the IEEE has the right to publish, distribute and reprint the Work in all forms and media.

(3)_____ _____

**Authorized Signature** **Date**

(Authors who are British or British Commonwealth Government employees should also sign line (1) above to indicate their acceptance of all terms other than the copyright transfer.)

**Return Fax to: +1 714 761 1784 Attn: Stephanie Kawada – WET ICE 2004**

# IEEE COPYRIGHT FORM *(continued)*

## RETAINED RIGHTS/TERMS AND CONDITIONS

1. Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.

2. Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.

3. Authors/employers may make limited distribution of all or portions of the Work prior to publication if they inform the IEEE in advance of the nature and extent of such limited distribution.

4. In the case of a Work performed under a U.S. Government contract or grant, the IEEE recognizes that the U.S. Government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract/grant so requires.

5. For all uses not covered by items 2, 3, and 4, authors/employers must request permission from the IEEE Intellectual Property Rights office to reproduce or authorize the reproduction of the Work or material extracted verbatim from the Work, including figures and tables.

6. Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use. The IEEE Intellectual Property Rights office must handle all such third-party requests.

## INFORMATION FOR AUTHORS

**IEEE Copyright Ownership**

It is the formal policy of the IEEE to own the copyrights to all copyrightable material in its technical publications and to the individual contributions contained therein, in order to protect the interests of the IEEE, its authors and their employers, and, at the same time, to facilitate the appropriate re-use of this material by others. The IEEE distributes its technical publications throughout the world and does so by various means such as hard copy, microfiche, microfilm, and electronic media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various compendiums, collective works, databases and similar publications.

**Author/Employer Rights**

If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IEEE assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

**Reprint/Republication Policy**

The IEEE requires that the consent of the first-named author and employer be sought as a condition to granting reprint or republication rights to others or for permitting use of a Work for promotion or marketing purposes.

**PLEASE DIRECT ALL QUESTIONS ABOUT THIS FORM TO:**
**Manager, IEEE Intellectual Property Rights Office, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1131.**
**Telephone +1 (732) 562-3966**

Jet Propulsion Laboratory
California Institute of Technology

4800 Oak Grove Drive
Pasadena, California 91109-8099

(818) 354-4321

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY

Authorization for Public Release and Transfer of Copyright

The contribution entitled **"Security Risks: Management and Mitigation in the Software Life Cycle" by D. Gilliam** submitted for publication in the **WET ICE 2004** has been cleared for public release by the Jet Propulsion Laboratory, California Institute of Technology. The copyright to the Contribution is transferred to: **The Institute of Electrical and Electronics Engineers (IEEE)** when the Contribution is published by the above-named publisher, with the following reservation of rights:

This Contribution was produced by (a) members(s) of the Jet Propulsion Laboratory, California Institute of Technology and is considered a work-for-hire. In accordance with the contract between the California Institute of Technology and the National Aeronautics and Space Administration, the United States Government and others acting on its behalf shall have, for Governmental purposes, a royalty-free, nonexclusive, irrevocable, world-wide license to publish, distribute, copy, exhibit and perform the work, in whole or in part, to authorize others to do so, to reproduce the final published and/or electronic form of the Contribution, to include the work on the NASA/JPL Technical Reports Server web site, and to prepare derivative works including, but not limited to, abstracts, lectures, lecture notes, press releases, reviews, textbooks, reprint books, and translations.

I, as an author, represent that I am authorized to sign for and on behalf of all authors, and that this agreement is made on behalf of all the authors.

D. Gilliam 06-04-04
(Signature) (Date)

Name: DAVID GILLIAM

AUTHORIZED REPRESENTATIVE

K. Lynn 6/4/04
(Signature) (Date)

Name: K. LYNN

Logistics & Technical Information Division
JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY

04-0778

# Security Risks: Management and Mitigation in the Software Life Cycle

David P. Gilliam
*Jet Propulsion Laboratory, California Institute of Technology*
david.p.gilliam@jpl.nasa.gov

## Abstract

*A formal approach to managing and mitigating security risks in the software life cycle is requisite to developing software that has a higher degree of assurance that it is free of security defects which pose risk to the computing environment and the organization. Due to its criticality, security should be integrated as a formal approach in the software life cycle. Both a software security checklist and assessment tools should be incorporated into this life cycle process and integrated with a security risk assessment and mitigation tool. The current research at JPL addresses these areas through the development of a Software Security Assessment Instrument (SSAI) and integrating it with a Defect Detection and Prevention (DDP) risk management tool.*

## 1. Introduction

Information Technology (IT) security, until recently, has had only minor attention in software engineering. With the growing number of system security defects being discovered and as the impact of malicious code grows, addressing security in the software life cycle is an increasingly crucial activity. There are several reasons for the number of vulnerabilities and security exposures in software. Two key issues are, 1) "The attitude today is that you can write any sloppy piece of code and the compiler will run diagnostics;" and the developer relies on the complier to indicate any errors, and 2) "The constant demand for novelty means that software is always in the bleeding-edge phase, when products are inherently less reliable." To these reasons should also be added lack of skilled developers and training, and lack of resources such as code analyzers that can check the security of software and systems. As McGraw points out, "Security is like fault tolerance, a system-wide emergent property that requires much advance planning and careful design." [1]

## 2. Current State of Software Security

There are several factors that need consideration when addressing security risk and risk management in the life cycle. These may include security requirements elicitation from the various stakeholders, the environment the software will run in, including hardware and other integrated software modules, expected life-time usage, and so on. These factors need consideration in addressing security in the software life cycle. The process of risk management can be aided by the use of tools. A security template for a risk management tool can aid in the process of determining what security risks need to be addressed, available mitigations and their effectiveness. Model checking and property-based testing have also shown to be effective instruments for reducing risks. As a result, JPL has developed a software security assessment instrument to aid in reducing vulnerability risks in software.

Modeling and property-based testing (PBT) are two techniques that can aid in reducing security risks. [2] As the complexity of software and requirements increases, software modeling becomes ever more difficult due to state space explosion. A model-based verification Flexible Modeling Framework (FMF) instrument is developed to help mitigate state space explosion. A Property-Based Tester (PBT) verifies code for the security properties specified in the requirements. Both the FMF and PBT can work together or be used as independent instruments. In addition, other tools that are under development are a security assessment checklist, training for software engineers and developers, and a vulnerability matrix that classifies vulnerabilities and identifies security property violations. These tools in conjunction with a security risk management instrument can reduce security vulnerabilities in the life cycle. The instrument has been funded by the NASA Independent Verification and Validation (IV&V) Center (Figure 1).

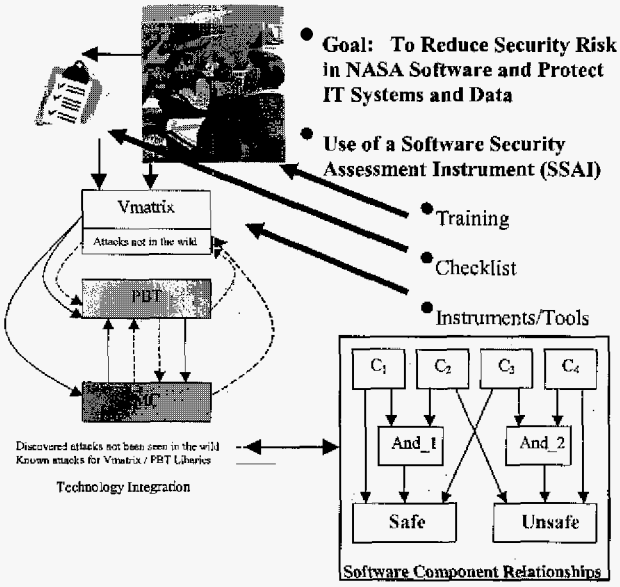


Figure 1: Software Security Assessment Instrument [11]

## 3. Related Work

Addressing security risk has taken a greater importance in the software life cycle. The Gartner Group points out that "IT assets that put an enterprise at risk must be identified through an IT risk assessment inventory that covers multiple domains in an organization." [3] Not directly included in their assessment is IT security in the life cycle. Other security risk management approaches also address enterprise security risk management from a system or site qualification perspective.[4] ISO9000 and the Capability Maturity Model Integration (CMMI) address the importance of managing risk, although they do not mention managing security risk. CMMI provides models for improvement and management of risk in the software development life cycle.[5,6] Likewise, the Software Engineering Institute (SEI), provides several publications and guides for an analysis team to conduct a risk evaluation for their organization.[7] The National Institute of Health's (NIH) Center for Information Technology (CIT) has addressed the problem of IT risk management by providing an application/system security plan template that identifies several types of security controls [8] as guidance in integrating security into the institutional processes and the project life cycle. It can be used in coordination with the security risk template and process discussed below.

While, software security has taken on an increasing importance in the life cycle, many approaches do not cover the relationship and integration of the life cycle and institutional risk management processes. Additionally, the process of retiring software and systems often is not addressed in security risk management and mitigation; nor is it addressed in vulnerability assessments. When software and systems are retired, undesired security exposures and vulnerabilities may be introduced if other systems dependent on communicating with them are left in an open state waiting for non-existent communication. This state can be exploited by a rogue system that masquerades as the retired system or by other means.

## 4. Establishing a Software Security Risk Management and Reduction Process

The planning and design for security begins with elicitation and specification of security requirements. However, these requirements must address identified needs based on the environment and customer needs. They must also be integrated with requirements for reliability and safety. Managing and mitigating, security reliability and safety risks are three foci that require being addressed collectively in the life cycle.

A risk assessment methodology is needed to aid and guide this process. The National Institute of Standards and Technology (NIST) "Risk Management Guide for Information Technology Systems," presents a nine step process to risk. [8, 9] A risk analysis methodology must quantify the cost of a risk occurrence, and the cost to mitigate it as used in Probability Risk Assessment. [10]

In application to IT security, risk is a function of the impact to a system when a vulnerability or exposure is exploited and the likelihood of its success, along with the frequency at which such events are perpetrated. Quantifying risk in these terms depends on the relative cost of the potential loss or disruption to the organization or the IT environment should the risk event occur. A simple algorithm to quantify IT security risk can be defined as: Risk = impact * likelihood * frequency. The frequency of an exploit being perpetrated is based on three factors: how easy it is to perpetrate an attack, the likelihood the attack will be successful, and the impact if it is successful.[11]

A key characteristic that makes security risk management an even more complex undertaking is that attacks having a greater potential for success and a greater propensity for damage will likely be perpetrated more frequently. Sophistication of attacks and exploits, and system complexity factors make it difficult to ascertain their risk. Consequently, risk management must be a persistent process as security risks will change over time as will the effectiveness and types of available mitigations.

Security risk is similar to other identified key risk areas like safety and reliability. These risks areas may impact each other as well and must be considered together when managing risk. A reliability risk may negatively impact security such as a software system that fails insecure. Likewise a security risk may also negatively impact safety such as a weak authentication process allows an unauthorized person to access a safety critical system such as a medical prescription database and modify data. Similarly, a mitigation for a security risk may positively impact safety such as a strong authentication scheme for such a system above. However, a mitigation like use of anti-virus (AV) software to prevent infection by malicious content may impact the reliability of a software system when the AV software scans the system for malicious content degrading processor and disk performance when the processor is needed for other real-time processes. [14]

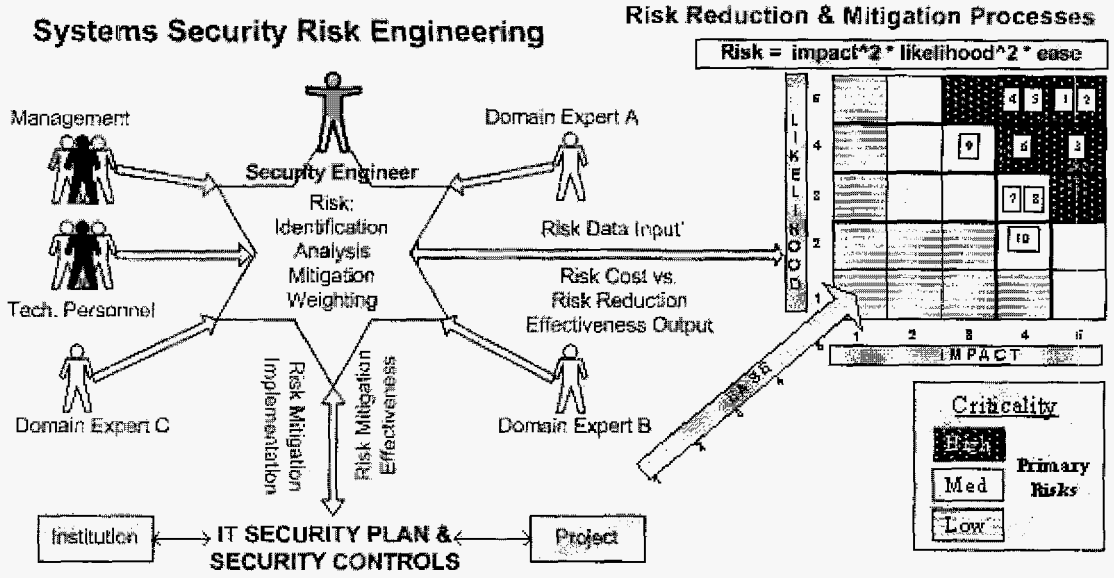

Figure 2: Software Security Assessment Instrument [14]

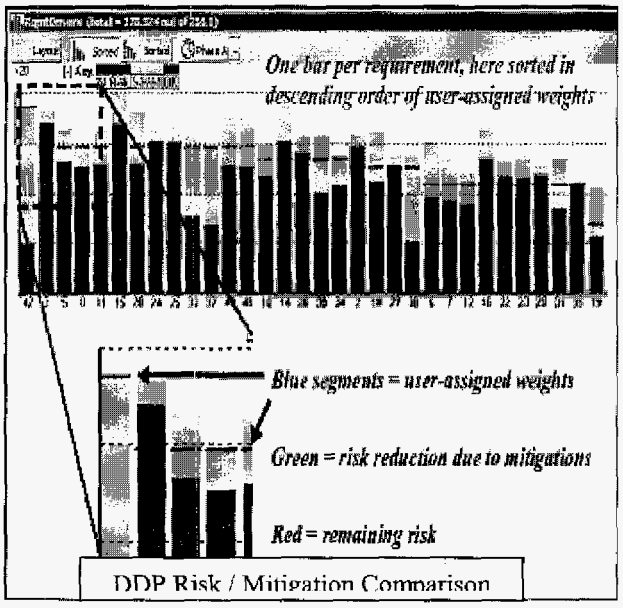

Figure.3: Risks in descending order of risk levels [12]

### 4.1 Assessing and Rating Security Risks – The Defect Detection and Prevention (DDP) Tool

Rating risks and potential mitigations is a difficult activity that is resource intensive. Automating as much of the process as possible is essential to managing risks. The process shown in Figure 2 involves domain experts to evaluate the risks and a security engineer to input the data into a database which can then be used by a risk management tool like DDP. "DDP explicitly represents risks, the objectives that risks threaten, and the mitigations available for risk reduction. By linking these concepts, DDP is able to represent and reason about the cost-effectiveness of risk reduction alternatives." [12] "The single most important aspect of the DDP approach is that it supports multiple experts [who] pool their knowledge" allowing them "to take the sum total of their pooled knowledge into account" for decisions." [13]

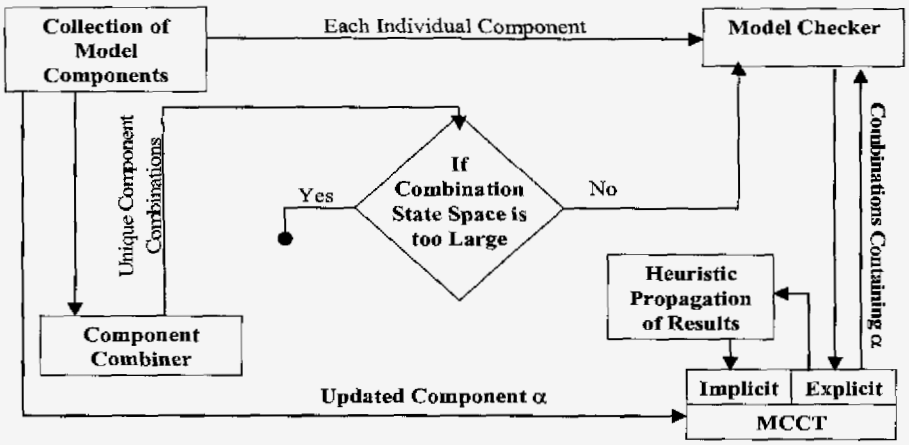


Figure.4: Illustration of the Flexible Modeling Framework process [20]

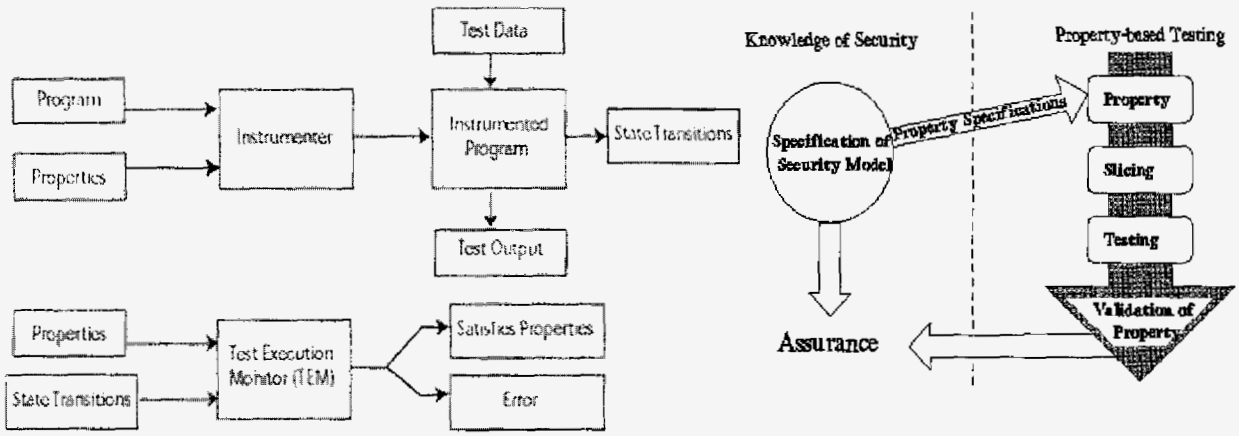


Figure 5: PBT Process and Model [20]

The goal is to evaluate relative security risks and mitigations. DDP can discover optimal solutions to risks and mitigations in terms of effectiveness and costs by iterating through the risks and their mitigations. DDP analyzes the data to find the optimal range for risk mitigations for a cost/benefit analysis. The tool provides the capability to compare risks and mitigations individually. DDP can then merge that data with the data for reliability and safety to analyze changes in risks and the impact of that one domain area may have on another (see Figure 3). [14]

### 4.2 Software Life Cycle Risk Mitigations

Security risk mitigations for the life cycle first require a controlled software engineering process that begins with requirements inception and follows through to decommissioning. A process for customer and stakeholder requirements specification will provide some mitigation for missed requirements and poor specifications. Mitigations can include Security Checklists (SC) and modeling. The requirements should address institutional requirements, institutional security risks from both internal and external threats.

An SC can provide system engineers direction in determining the scope and environment of security settings both for software and the network systems and environment in which the system will run. The SC should include a process for identifying stakeholders, how the system will be used, what its purpose is, etc. [15] These items should be specified in a Concept of Operations document that guide the system engineer in developing the requirements specifications. Security requirements may include authentication, role access controls, secure communications, and fail-secure requirements. [16] After specification, the requirements can be modeled using model-based verification techniques. [17] Model checking "can identify vulnerabilities and undesired exposures in software. These often arise from a number of development factors that can often

be traced to poor software development practices, new modes of attacks in the network security arena, unsafe configurations, and unsafe interaction between systems and/or their components." [18]

One of the problem areas facing modeling techniques is state-space explosion where the model grows rapidly as the complexity of the system increases. [19] To address this problem, research at JPL has led to the development of the Flexible Modeling Framework (FMF). [20] The FMF builds up a model of the system from components which are first verified for correctness using the SPIN model checker. They are then combined with other modeled components and re-verified for correctness. In this approach security properties of a system can be verified while maintaining system fidelity and reducing state-space explosion (Figure 4). [17]

An additional advantage of the FMF is in the ability to re-use the model during the maintenance phase of the life cycle where many security violations arise. Additionally, there is a potential for component and model reuse which can reduce the cost of model development on other software systems that may use these components. These security property specifications should then be tested during the development, test and verification, and the maintenance phases of the life cycle.

Another tool being developed at JPL is a property-based testing (PBT) instrument (Figure 5). This instrument can use the security properties specified and used by the FMF. The PBT technique is code insertion as comments. The footprint adds negligible size to the code. [18] The PBT aids in security risk mitigation by providing another element to the life cycle process.

The use of the software security assessment instruments in the development and maintenance life cycles reduces over-all risk to the organization and to those who use the software. Used together, they provide a much higher level of assurance that the software will be free of security defects. Used independently, they can still provide a higher level of assurance, but will not be as effective.

In integrating these tools and instruments in the risk analysis as risk mitigations, their value increases with the complexity of the code and the extent of maintenance that will be required, especially if the software, while still in the development phases, will be used by the organization while it attempts to detect and fix defects as part of its testing process and environment. It is imperative that these factors be integrated into a security risk analysis if an organization is to analyze its risk profile.

## 5. Conclusion

The software security assessment instrument developed at JPL can be used collectively or individually as needed. Figure 1 shows various pieces of the instrument can work in concert to provide a greater level of security assurance. Coupling the security assessment instrument with DDP provides further benefit for being able to determine the level of risk and the effectiveness of the mitigations of the tools in a changing threat scenario. As a security risk mitigation instrument, both the FMF and PBT provide significant gains during the maintenance phase of the life cycle when software upgrades or extensions to the system are released. This gain can be modeled with the DDP tool giving management a better view of the potential payback for use of these instruments during the early stages of the life cycle when it is easier to develop and check the model, and test to verify that security properties are not violated in the development phase. When the software system enters the maintenance phase, the FMF model can be modified and re-used to re-verify connecting components where changes are made. and the PBT tool can be used to assess potential security property violations in the code.

By addressing security risks in the software life cycle through formal approaches, the security of systems and organizations will improve greatly. By making it a persistent process, the security of software will attain a higher level of assurance and provide the capability of responding to new emerging threats with a greater degree of trust in the software. It is essential that security be addressed in the life cycle to reduce vulnerabilities and unwanted exposures in software and systems. The approach must be an inclusive formal approach that begins with requirements elicitation and specification and ends with the decommissioning or retirement of the software system.

## 6. Acknowledgements

The research described in this paper is being carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## 7. References

[1] McGraw, G., "Software Risk Management for Security", Citigal White Paper, 1999, available at

http://www.cigital.com/whitepapers/. See also, Hoglund, G., McGraw, G., Exploiting Software : How to Break Code, Pearson Higher Education, February, 2004.

[2] Gilliam, D.G., and Powell, J.D., "Integrating a Flexible Modeling Framework with a Security Assessment Instrument to Reduce Software Security Risk," Proc. of the 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), June 2002, Pittsburgh, pp. 153-158.

[3] Witty, R., "Successful Elements of an Information Security Risk Management Program," Gartner Symposium ITxpo, U.S. Symposium/ITxpo, Orlando, Florida, 6-11 October 2002.

[4] RiskWatch, "Security Risk Management (SRM) software solutions for government and industry," http://www.riskwatch.com/, accessed 02-10-04,. See also: Stoneburner G., Goguen, A., and Feringa, A., "Risk Management for Information Technology Systems," The National Institute of Standards and Technology Special Publication 800-30, 2001 and SEI, Carnegie Mellon University Software Engineering Institute, "OCTAVE Method," November 2003, available at: http://www.cert.org/octave/methods.html.

[5] ISO, International Organization for Standardization, ISO 9000:2000 family, Quality Management Systems, http://www.iso.ch/iso/en/iso9000-14000/iso9000/selection_use/iso9000family.html , accessed 03-19-2004.

[6] Carnegie Mellon University (CMU) Software Engineering Institute (SEI) Capability Maturity Model® Integration (CMMI$^{SM}$), available at http://www.sei.cmu.edu/cmmi/general/.

[7] SEI, Carnegie Mellon University Software Engineering Institute, "OCTAVE Method," http://www.cert.org/octave/methods.html., accessed 03-11-2004.

[8] NIH CIT (National Institute of Health, Center for Information Technology), "NIH Application/System Security Plan Template for Major Applications and General Support Systems," 1994.

[9] NASA CRM Resource Center website http://www.crm.nasa.gov/knowledge/default.html , accessed 03-05-2004.

[10] Stamatelatos, M.G., "Risk Assessment and Management, Tools and Applications," PowerPoint Presentation, available at http://www.crm.nasa.gov/papers/presentation_1.pdf , accessed 03-20-04. See Hassett, M.J., Stewart, D., and Stewart, D.A., Probability for Risk Management, Actex Pubns Inc., May 1999.

[11] Gilliam, D.P., "Managing Information Technology Security Risk," International Symposium On Software Security, Tokyo Institute of Technology, Tokyo, Japan, November 2003.

[12] Feather, M.S., Cornford, S. L., and Moran, K., "Risk-Based Analysis And Decision Making In Multi-Disciplinary Environments," Proc. of IMECE'03 2003 ASME International Mechanical Engineering Congress & Exposition Washington, D.C., November 16–21, 2003.

[13] Cornford, S. L., Feather, M.S., Dunphy, J., Salcedo, J., and Menzies, T., "Optimizing Spacecraft Design – Optimization Engine Development: Progress and Plans," IEEE Aerospace Conference, March 2003, (available on the web at: http://ddptool.jpl.nasa.gov).

[14] Gilliam, D.G., Feather, M.S., "System Security Engineering," to appear in the International Consortium on System Engineering (INCOSE) 2004, that describes the use of a DDP security template for security risk.

[15] Gilliam, D.G., Wolfe, T.L., Sherif., J.S., "Software Security Checklist for the Software Life Cycle," Proc. of the 11th IEEE WETICE Workshop, June 2003, Linz, Austria, pp. 243-248.

[16] Bishop, M., Computer Security: Art and Science, Addison-Wesley Pub Co., December, 2002.

[17] Wen, W., and Mizoguchi, F., "Model checking Security Protocols: A Case Study Using SPIN, IMC," Technical Report, November, 1998.

[18] Powell, J.D., Gilliam, D.P., "Component Based Model Checking," Proceedings of the 6th World Conference on Integrated Design and Process Technology, June 23-28, Pasadena CA, p. 66.

[19] Holzmann, G., The SPIN Model Checker : Primer and Reference Manual, Pearson Educational, September 4, 2003. See also, Ammann, P.E., Black, P.E., and Majurski, W., "Using Model Checking to Generate Test Specifications," 2nd International Conference on Formal Engineering Methods (1998) pp. 46-54.

[20] Gilliam, D.P., Powell, J.D., Bishop, M., "Reducing Software Security Risk Through an Integrated Approach", IEEE Goddard 28th Software Engineering Workshop, Greenbelt, MD, Dec. 1-4, 2003, pp. 36-42.

04-0778

**Return Fax to: +1 714 761 1784 Attn: Stephanie Kawada – WET ICE 2004**

# IEEE COPYRIGHT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

**TITLE OF PAPER/ARTICLE/REPORT/PRESENTATION/SPEECH (hereinafter, "the Work"):**

**COMPLETE LIST OF AUTHORS:**

**IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book): WET ICE 2004**

## Copyright Transfer

The undersigned hereby assigns to the Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to the above Work, and any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work. The undersigned hereby warrants that the Work is original and that he/she is the author of the Work; to the extent the Work incorporates text passages, figures, data or other material from the works of others, the undersigned has obtained any necessary permissions. **See reverse side for Retained Rights and other Terms and Conditions.**

## Author Responsibilities

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements of IEEE Policy 6.3, including provisions covering originality, authorship, author responsibilities and author misconduct. The full policy may be viewed at **http://www.ieee.org/about/whatis/policies/p6-3.html**. Authors are advised especially of IEEE Policy 6.3.1A(l): "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it."

## General Terms

- The undersigned represents that he/she has the power and authority to make and execute this assignment.
- The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
- In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall become null and void and all materials embodying the Work submitted to the IEEE will be destroyed.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.

(1)_____ _____

**Author/Authorized Agent for Joint Authors** **Date**

## U.S. Government Employee Certification (where applicable)

This will certify that all authors of the Work are U.S. government employees and prepared the Work on a subject within the scope of their official duties. As such, the Work is not subject to U.S. copyright protection.

(2)_____ _____

**Authorized Signature** **Date**

(Authors who are U.S. government employees should also sign signature line (1) above to enable the IEEE to claim and protect its copyright in international jurisdictions.)

## Crown Copyright Certification (where applicable)

This will certify that all authors of the Work are employees of the British or British Commonwealth Government and prepared the Work in connection with their official duties. As such, the Work is subject to Crown Copyright and is not assigned to the IEEE as set forth in the first sentence of the Copyright Transfer Section above. The undersigned acknowledges, however, that the IEEE has the right to publish, distribute and reprint the Work in all forms and media.

(3)_____ _____

**Authorized Signature** **Date**

(Authors who are British or British Commonwealth Government employees should also sign line (1) above to indicate their acceptance of all terms other than the copyright transfer.)

**Return Fax to: +1 714 761 1784 Attn: Stephanie Kawada – WET ICE 2004**

# IEEE COPYRIGHT FORM *(continued)*

## RETAINED RIGHTS/TERMS AND CONDITIONS

1. Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.

2. Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.

3. Authors/employers may make limited distribution of all or portions of the Work prior to publication if they inform the IEEE in advance of the nature and extent of such limited distribution.

4. In the case of a Work performed under a U.S. Government contract or grant, the IEEE recognizes that the U.S. Government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract/grant so requires.

5. For all uses not covered by items 2, 3, and 4, authors/employers must request permission from the IEEE Intellectual Property Rights office to reproduce or authorize the reproduction of the Work or material extracted verbatim from the Work, including figures and tables.

6. Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use. The IEEE Intellectual Property Rights office must handle all such third-party requests.

## INFORMATION FOR AUTHORS

**IEEE Copyright Ownership**

It is the formal policy of the IEEE to own the copyrights to all copyrightable material in its technical publications and to the individual contributions contained therein, in order to protect the interests of the IEEE, its authors and their employers, and, at the same time, to facilitate the appropriate re-use of this material by others. The IEEE distributes its technical publications throughout the world and does so by various means such as hard copy, microfiche, microfilm, and electronic media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various compendiums, collective works, databases and similar publications.

**Author/Employer Rights**

If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IEEE assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

**Reprint/Republication Policy**

The IEEE requires that the consent of the first-named author and employer be sought as a condition to granting reprint or republication rights to others or for permitting use of a Work for promotion or marketing purposes.

**PLEASE DIRECT ALL QUESTIONS ABOUT THIS FORM TO:**
**Manager, IEEE Intellectual Property Rights Office, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1131.**
**Telephone +1 (732) 562-3966**

Jet Propulsion Laboratory
California Institute of Technology

4800 Oak Grove Drive
Pasadena, California 91109-8099

(818) 354-4321

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY

Authorization for Public Release and Transfer of Copyright

The contribution entitled **"Security Risks: Management and Mitigation in the Software Life Cycle" by D. Gilliam** submitted for publication in the **WET ICE 2004** has been cleared for public release by the Jet Propulsion Laboratory, California Institute of Technology. The copyright to the Contribution is transferred to: **The Institute of Electrical and Electronics Engineers (IEEE)** when the Contribution is published by the above-named publisher, with the following reservation of rights:

This Contribution was produced by (a) members(s) of the Jet Propulsion Laboratory, California Institute of Technology and is considered a work-for-hire. In accordance with the contract between the California Institute of Technology and the National Aeronautics and Space Administration, the United States Government and others acting on its behalf shall have, for Governmental purposes, a royalty-free, nonexclusive, irrevocable, world-wide license to publish, distribute, copy, exhibit and perform the work, in whole or in part, to authorize others to do so, to reproduce the final published and/or electronic form of the Contribution, to include the work on the NASA/JPL Technical Reports Server web site, and to prepare derivative works including, but not limited to, abstracts, lectures, lecture notes, press releases, reviews, textbooks, reprint books, and translations.

I, as an author, represent that I am authorized to sign for and on behalf of all authors, and that this agreement is made on behalf of all the authors.

D. Gilliam 06-04-04
(Signature) (Date)

Name: DAVID GILLIAM

AUTHORIZED REPRESENTATIVE

K. Lynn 6/4/04
(Signature) (Date)

Name: K. LYNN

Logistics & Technical Information Division
JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY