



## Professional Training in Software Engineering: A Critical Need in the United States

## Formación Profesional en Ingeniería de Software: Una Necesidad Crítica en los Estados Unidos

Jennifer Waldrow

The Pennsylvania State University. [jenwal\(AT\)psu.edu](mailto:jenwal(AT)psu.edu)

### INFORMACIÓN DEL ARTÍCULO

*Tipo de artículo:* Reflexión.

*Historia del artículo*

Recibido: 13-01-2012

Correcciones: 21-05-2012

Aceptado: 25-05-2012

*Categories and Subject Descriptors*

K.3.2 [Computers and Education]:  
Computer and Information Science  
Education – Curriculum.

*General Terms*

Software Engineering, Computer  
Science.

*Keywords*

Software Engineering, Computer  
Science, education, complexity.

*Palabras clave*

Ingeniería de Software, Ciencias  
Computacionales, formación,  
complejidad.

### ABSTRACT

The software is related to almost every aspect of daily life: manufacturing, banking, travel, communications, defense, medicine, research, government, education, entertainment, law ... Is an essential part of military systems and is used in all civilian sectors, including safety and mission critical. Moreover, the complexity of many of these systems has increased exponentially in recent decades and the software has become an essential component for all of them. Unfortunately, the "systems of higher education", in almost all countries have not kept pace with these changes. The current science and engineering programs, both undergraduate and graduate, they need to incorporate more training in Software Engineering. It is especially true in areas such as aerospace engineering, because these systems are highly dependent on computer, information, communications and software. This article presents an analysis of the current situation of the United States in what has to do with software engineering training that receive and require the aerospace engineers.

### RESUMEN

El software se relaciona con casi todos los aspectos de la vida cotidiana: manufactura, banca, viajes, comunicaciones, defensa, medicina, investigación, gobierno, educación, entretenimiento, leyes,... Es una parte esencial de los sistemas militares y se utiliza en todos los sectores civiles, incluyendo a los de seguridad y los de misión crítica. Además, la complejidad de muchos de estos sistemas se ha incrementado exponencialmente en las últimas décadas y el software se ha convertido en un componente esencial para todos ellos. Por desgracia, los "sistemas de educación superior", en casi todos los países, no le han seguido el paso a estos cambios. Los actuales programas de ciencia e ingeniería, tanto de pregrado como de posgrado, necesitan incorporar más capacitación en Ingeniería de Software. Lo que es especialmente cierto en áreas como la Ingeniería Aeroespacial, porque estos sistemas dependen en gran medida de la computación, la información, las comunicaciones y el software. En este artículo se presenta un análisis a la situación actual de los Estados Unidos en lo que tiene que ver con la formación en Ingeniería de Software que reciben y requieren los ingenieros aeroespaciales.

© 2012 IAI. All rights reserved.

### 1. INTRODUCCIÓN

El software está en todas partes, desde los teléfonos celulares hasta los grandes sistemas militares y, de acuerdo con la National Academy of Science [1], el software no es más que un artículo de comercialización básico que, de hecho, incorpora en sí a la función productiva de la economía. El National Institute of Standards and Technology NIST estima que los errores del software le cuestan a la economía de los EE.UU. \$59,5 billones al año y que las ventas de productos software representan \$180 billones [2], pero la formación en Ingeniería de Software no recibe la atención que merece, a pesar de que es de vital importancia para su economía. Las cuestiones relacionadas con la Ingeniería de software como disciplina y los debates en los últimos años no son nuevos y se describen en [3-5].

La lista de los problemas causados por productos software se incrementa cada año. Históricamente se pueden mencionar: el Ariane 5 [6, 7], el Federal Bureau of Investigation Virtual Case File System [8], el Federal Aviation Administration Advanced Automation System [7, 9], el California Department of Motor Vehicle system, el sistema de reservas de American Airlines, el sistema complejo del avión F-22, y así se podrían mencionar muchos más [7, 10]. Sólo en Estados Unidos, estos "desastres" cuestan miles de millones dólares cada año y la sensación es que esto tiende a empeorar debido a que los sistemas futuros serán más complejos. Boeing invirtió cerca de 800 millones de dólares en el software del 777 y se calcula que para el 787 fue cinco veces más [11]. Los sistemas aeroespaciales también incluyen ciertos niveles de autonomía, acompañados de un nivel totalmente

nuevo de complejidad en el software. Para ayudar a prevenir futuros desastres, se requieren más ingenieros de software formados en programas profesionales rigurosos que estén al nivel de cualquier programa de ingeniería. Países como Estados Unidos no pueden esperar hasta que se presenten los desastres con pérdidas de vidas o económicos para tomar medidas correctivas. Ahora es el momento para que se tomen las medidas preventivas necesarias, pero actualmente no existen suficientes ingenieros de software, ni programas profesionales que los formen adecuadamente. Se necesita formar a muchos más ingenieros para el futuro cercano, especialmente si se considera el alto número que se jubilará en los próximos diez años [12]. Por ejemplo, en 2005, la edad promedio de los ingenieros de software era de 54 años [13], además, cerca del 26% de los profesionales en esta área será elegible para jubilación en el año 2013 [14].

## 2. UN EJEMPLO DE LA CRISIS

La industria aeroespacial estadounidense provee cerca de 900 billones de dólares como actividad económica, representa más del 15% del producto interno bruto y es soportada por más de 15 millones de puestos de trabajo de alta calidad [14]. Los sistemas aeroespaciales dependen en gran medida del software, al que se ha llamado su talón de Aquiles. Existen numerosas anécdotas y ejemplos que ilustran la importancia de la informática y del software de esta industria, por ejemplo:

- El Boeing 777 cuenta con 1.280 procesadores a bordo que utilizan más de cuatro millones de líneas de código, de las cuales el 99,5% es en Ada [15, 16].
- El F-22 tiene más de dos millones de líneas de código de a bordo, entre el 80 y el 85% está en Ada [17].
- Algunos helicópteros Blackhawk tienen casi 2.000 kilos de cables de conexión en todos los equipos y sensores.
- El cableado es a menudo más complejo y más difícil de diseñar que la estructura del avión.
- Algunos aviones no pueden volar sin sus computadores de a bordo, por ejemplo el F-16 y el F-117.
- El sistema de control aéreo se basa principalmente en computadores, software y comunicaciones.
- Los robots y las naves espaciales interplanetarias realizan hazañas increíbles, a menudo en ambientes extremos.
- Los vehículos no tripulados, inteligentes y autónomos serán mucho menos deterministas que los sistemas actuales.
- Los computadores también son importantes en el diseño y en el análisis de los sistemas aeroespaciales. A menudo esto significa el uso de sistemas de cómputo paralelo de alto rendimiento.
- Los sistemas de comunicación son de vital importancia para las aeronaves y las naves espaciales, lo que incluye redes de computadoras a bordo, en tierra y para otros vehículos aeroespaciales.
- Los aviones y naves espaciales modernos rara vez trabajan solos, por lo general hacen parte de un sistema de sistemas.

Una manera de medir la necesidad de ingenieros de software en el campo aeroespacial es investigar las oportunidades de empleo existentes. En 2006, una revisión al sitio Lockheed-Martin Corporation Web mostraba que tenían 536 vacantes de empleo para graduados recientes, incluyendo 68 en ingeniería de software y cuatro en ingeniería aeroespacial. La mayoría de los puestos de trabajo de ingeniería aeroespacial se ofrecen para ingenieros estructurales capaces de realizar análisis de elementos finitos. Cabe señalar que no se contratan ingenieros aeroespaciales en otras áreas, por ejemplo, los expertos en control aeroespacial a veces aparecen listados en los sistemas embebidos.

La página de empleo de la Boeing ofreció resultados similares. Tenían 298 ofertas de trabajo relacionadas con software. Sólo había tres puestos de trabajo que mencionaban la aerodinámica, ninguno de los cuales era en realidad puesto de trabajo para ingenieros aerodinámicos. Al buscar empleo para ingenieros aeroespaciales la página devolvió una lista de seis columnas en ingeniería estructural. Esta es probablemente una indicación de que los programas formativos en ingeniería aeroespacial se concentran demasiado en la física aplicada a la ingeniería aeroespacial y no lo suficiente en la informática y el software. Se debe trabajar con la industria y el gobierno para redefinir la ingeniería aeroespacial. Es necesario formar estudiantes capaces de diseñar y de construir los sistemas aeroespaciales nuevos que se necesitarán en el futuro y que estarán dominados por los sistemas de computación, las redes y la información.

## 3. EL DOMINIO DE LA INGENIERÍA DE SOFTWARE

El Institute of Electrical and Electronics Engineers IEEE define Ingeniería de Software como la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento de software [3]. Un buen resumen de Ingeniería de Software se puede encontrar en [18].

Los sistemas software son algunas de las cosas más complicadas que los seres humanos jamás han creado. Para diseñarlas y construirlas, se deben seguir los procesos y procedimientos propios de otras disciplinas ingenieriles [6, 19]. En primer lugar, definir cuidadosamente los requisitos para luego desarrollar la arquitectura del sistema. Una vez que se definen los requisitos y la arquitectura, se puede comenzar el desarrollo de código. Luego, el código necesita verificación, validación y pruebas. Existen diversas formas de llevar a cabo todos estos pasos, que se relacionan con el tipo de modelo de ciclo de vida utilizado y el tipo de sistema desarrollado. Además, es necesario considerar cómo estimar los costos, cómo interactuar con las personas y cómo monitorear las responsabilidades éticas del equipo. Este proceso no es diferente del necesario para diseñar y construir cualquier sistema complejo, por ejemplo, puentes, aviones y equipo informático. El desarrollo de código, a veces mal llamado programación, puede ser una parte muy pequeña de todo el proceso [6].

La mayoría de ingenieros y científicos no aprecian o no entienden la Ingeniería de Software; incluso los estudiantes universitarios piensan que pueden hacer software luego de aprender los conceptos básicos de la sintaxis de un lenguaje y, frecuentemente, la definen como “programación”. Esto es como comparar a la ingeniería civil con el vertido de hormigón. Muchas personas pueden verter concreto, pero son pocos los ingenieros civiles que pueden inspirar técnicamente para construir obras maestras grandes. Del mismo modo, muchas personas pueden “programar”, pero pocas pueden “desarrollar” grandes obras maestras de software. No es raro escuchar discusiones acerca de las ventajas o de los inconvenientes de los diferentes lenguajes de programación, a pesar de que quienes participan no conocen bien la diversidad que existe. A menudo, simplemente les gusta el lenguaje que aprendieron y no aprecian ni comprenden los demás. Estos conceptos erróneos son especialmente evidentes en las discusiones sobre Ada [20], que probablemente siga siendo el mejor lenguaje que se pueda utilizar para desarrollar sistemas de misión crítica o de seguridad. En realidad, las personas que desarrollan código sin un apropiado enfoque de Ingeniería de Software no son más que piratas informáticos. Por supuesto, los desarrolladores son una parte esencial del equipo de trabajo y los talentosos son muy raros y extremadamente valiosos.

#### 4. LA FORMACIÓN EN INGENIERÍA DE SOFTWARE

Tanto la economía como la defensa nacional de los EE.UU. depende del software, pero muchos de estos grandes sistemas de software están siendo desarrollados por personas que nunca han sido formalmente capacitados en Ingeniería. Si bien existen algunos ingenieros de software autodidactas muy talentosos, no se debe confiar en que la mayoría lo sea. Si nunca se construye un avión moderno sin ingenieros aeroespaciales y si nunca se construyen puentes o edificios sin ingenieros civiles, entonces ¿por qué se desarrollan grandes sistemas software sin ingenieros formalmente entrenados y certificados profesionalmente?

No hace mucho, el Dr. John Knight, profesor de la Universidad de Virginia, contrastaba la Ingeniería de Software con otras disciplinas ingenieriles [21]. Habló de cómo las catedrales de 1000 años de antigüedad fueron construidas usando la mejor tecnología de la ingeniería civil de la época y cómo estos edificios siguen en pie. La ingeniería civil se ha desarrollado enormemente con el tiempo y ahora se tienen enormes rascacielos y puentes espectaculares. Esto no sería posible sin un buen sistema formativo en ingeniería civil, programas de investigación y asesorías. Analogías similares se pueden extraer de otras disciplinas de ingeniería. En un millar de años a partir de ahora, las personas se maravillarán con los hitos de la ingeniería aeroespacial actual, como el Wright Flyer, el Blackbird SR-71 y el programa Apolo. Todos fueron grandes proyectos de ingeniería en su día y entonces estarán en museos. ¿Habrá alguna catedral de software para maravillarse dentro de 1.000 años? ¿O las generaciones futuras nos verán sólo como piratas informáticos?

#### 4.1 La ciencia tradicional y la formación en ingeniería

Muchos de los estudiantes que se gradúan en los programas de ciencia y de ingeniería americanos, con el tiempo trabajarán en el desarrollo de software. Desafortunadamente, la mayoría de ellos han recibido poca o ninguna formación en software. Por ejemplo, el 24% de los graduados en física estarán trabajando en software entre cinco y ocho años después de graduarse [22] y, la mayoría, probablemente no recibió capacitación en Ingeniería de Software en la universidad. Otros graduados en ciencias, incluso fuera de la ingeniería, con el tiempo también podrán trabajar en el desarrollo de software. Sería muy beneficioso para estos estudiantes poder conocer más acerca de la Ingeniería de Software antes de graduarse; necesitan más que un curso en programación en el primer año. Este es el caso de los pregraduados en casi todas las ciencias tradicionales y de ingeniería.

Tampoco es válido suponer que los egresados de Ciencias Computacionales son Ingenieros de Software. Es bastante fácil graduarse de uno de estos programas con muy poca formación en desarrollo de software. Knight y Leveson [23] describen la necesidad de más programas de formación en Ciencias Computacionales y de ingeniería computacional y avocan por más programas de Ingeniería de Software.

La necesidad de una buena formación en software es especialmente crítica en los programas de ingeniería aeroespacial. Estos ingenieros siempre se han enorgullecido de ser integradores de sistemas, pero para ello debe tener cierta comprensión del sistema aeroespacial completo que se está desarrollando. En los aviones de combate modernos, los componentes electrónicos representan aproximadamente el 10% de su peso y el 33% del costo [24]. Así que si los ingenieros no están bien formados en computación, redes, sensores y software no podrán comprender el sistema completo, a menos que el sistema tenga 60 años de antigüedad. Los estudiantes de hoy necesitan ser formados para que desarrollen la próxima generación de sistemas aeroespaciales, no viejos aviones y naves espaciales. Los sistemas aeroespaciales siempre han utilizado la última tecnología para lograr su increíble rendimiento. Los sistemas actuales y los futuros dependen en gran medida de los computadores y el software y los estudiantes necesitan conocerlos. Los ingenieros aeroespaciales son necesarios como integradores de sistemas, pero esto sólo es posible si tienen algún conocimiento del sistema completo, incluyendo a la informática y al software.

Hoy en día esto va más allá de la aviación a bordo porque las aeronaves y naves espaciales modernas casi siempre están ligadas a otros sistemas, pero la aviación es una gran parte de los sistemas aeroespaciales. El poder de procesamiento y de memoria de los computadores se han incrementado de manera exponencial en los aviones militares desde 1960 [25]. El F-106 tenía menos de 20 KB de memoria, mientras que el Joint Strike Fighter JSF puede tener más de 2 GB. La aviación podría representar el 40% del costo de la JSF. Este reporte también señala

que el contenido de software en estos sistemas ha aumentado de manera espectacular y que se necesitan más ingenieros de software.

La informática y el software son partes integrales de la ingeniería aeroespacial. Tradicionalmente, esta ingeniería se ha construido sobre cuatro pilares tecnológicos [26]: aerodinámica, estructuras, propulsión y dinámica y control, que se ven reflejados en los programas respectivos. Estas disciplinas fueron importantes para los hermanos Wright y para todos los sistemas aeroespaciales desde entonces; sin embargo, la ingeniería aeroespacial moderna debe incluir cinco pilares. Estas áreas son [27]: aerodinámica, estructuras y materiales, propulsión, dinámica y control e informática y software. Los sistemas aeroespaciales actuales y futuros son y serán diseñados utilizando computación; tienen computadores a bordo y deben comunicarse con otros vehículos y computadores. La informática —incluido el procesamiento, las redes y el almacenamiento de datos— y el software son elementos esenciales de esta ingeniería y conforman el quinto pilar. Además, este pilar podría ser el más importante aunque es mucho menos maduro que los otros cuatro. Los programas formativos de esta ingeniería tienen un fuerte énfasis en la física aplicada —por ejemplo, dinámica de fluidos, dinámica estructural, dinámica, combustión y propulsión— e, históricamente, había buenas razones para ello, pero no ya no es posible seguir desatendiendo la investigación y las necesidades formativas en informática y software.

Mientras que la informática y el software son cruciales para los sistemas aeroespaciales, los programas formativos en la ingeniería por lo general no lo reflejan. La mayoría de ellos requieren alrededor de 40 cursos durante un período de cuatro años, pero los estudiantes suelen tomar sólo un curso relacionado con software —un curso de programación de primer año—. Además, usualmente no requieren aprender acerca de aviónica, sistemas embebidos, redes, sensores, o hardware de computador. Esta tendencia también se presenta en los posgrados, en los que en los exámenes de ingreso y planes de estudio rara vez se incluye informática, software, o aviónica. A menudo se aplican principalmente sobre programas de física.

Pennsylvania State University ha estado trabajando para modernizar sus planes de estudio en ingeniería aeroespacial [28] y ahora ofrece cursos de alto nivel en programación avanzada de computadores y de Ingeniería de Software. También cuenta con un nuevo curso en Sistema de Posicionamiento Global y, desde 2006, los estudiantes de ingeniería aeroespacial tienen que tomar el curso de Ingeniería de Software o un curso de circuitos electrónicos. Idealmente, deben tomar ambos y también estar expuestos a la ingeniería de sistemas, sistemas embebidos, redes, sistemas de información, sensores y software. Estos temas adicionales pueden ser cubiertos en cursos electivos técnicos o en cursos de posgrado.

También hay que señalar que es difícil formar un ingeniero en todo lo que necesita profesionalmente en

cuatro años. De hecho, la U.S. National Academy of Engineering [29] recomienda que la licenciatura en ciencias sea reconocida como un grado de pre-ingeniería. Los científicos e ingenieros necesitan continuar formándose durante toda su vida para ser eficaces. Además, muchos puestos de ingeniería aeroespacial requieren un grado de maestría, lo que le permite al estudiante concentrarse en un área en particular. Una excelente combinación sería obtener una licenciatura en ciencias de ingeniería aeroespacial y luego una maestría o doctorado en Ingeniería de Software o de sistemas. Estos graduados serían muy valiosos. Otra posibilidad sería ofrecer especializaciones en Ingeniería de Software. Penn State tiene un programa de estos en Ciencias Computacionales [30], lo que atrae a estudiantes de una amplia variedad de departamentos de ciencias e ingeniería. Un programa similar se puede crear para la Ingeniería de Software o la ingeniería de sistemas.

#### **4.2 Pregrados en Ingeniería de Software**

Como se dijo anteriormente, los programas existentes en ciencia e ingeniería necesitan incluir más informática y software en sus planes de estudio, pero también se necesitan programas de pregrado en Ingeniería de Software. Estos programas, sin embargo, necesitan incluir suficiente ciencia e ingeniería en sus currículos; por ejemplo, física, matemáticas, matemática discreta, lógica y abstracción y sistemas embebidos y no deben tener énfasis excesivo en gestión, negocios y procesos. ACM, IEEE y la National Science Foundation han desarrollado muy buenos planes de estudios de pregrado en Ingeniería de Software [3].

Actualmente en los Estados Unidos, sólo existen 10 programas de pregrado acreditados en Ingeniería de Software [31], mientras que hay 67 programas en ingeniería aeroespacial. El país necesita muchos más programas en la primera, algo que tiene que suceder pronto porque se necesitan años para iniciar nuevos programas y para que los estudiantes se gradúen, además, porque tiene una fuerza laboral que envejece. Algunas empresas del sector aeroespacial y de defensa estiman que el 40% de su fuerza laboral se jubilará en los próximos cinco años [12]. De acuerdo con el Wall Street Journal, organizaciones como la National Aeronautics and Space Administration tiene más ingenieros mayores de 60 años que menores de 30. Para estos pregrados en Ingeniería de Software existentes se ofrecen 109 maestrías y 40 doctorados, pero pocos de ellos, de pregrado o de posgrado, se ofrecen en las universidades de investigación más importantes. Además, sólo algunos existen en las universidades incluidas entre las 25 más importantes del ranking del U.S. News and World Report. La mayoría de estos programas se encuentran en escuelas relativamente pequeñas, tal vez porque son capaces de reaccionar más rápidamente a las necesidades de la industria y de la sociedad.

Por desgracia, en el mundo académico el cambio se produce muy lentamente, porque hay pocos incentivos para cambiar. La financiación gubernamental puede y debe ser usada para ayudar a acelerar estos cambios. Los



Líderes de la industria tienen que involucrarse y también demandar el cambio. Es necesario que haya prácticas y tutorías disponibles, lo mismo que una necesidad por la formación continua. En los laboratorios del gobierno y en la industria, existe gran necesidad de capacitación en Ingeniería de Software para su fuerza de trabajo existente. Otro asunto es la cuestión de la certificación profesional en Ingeniería de Software. El IEEE ha desarrollado un excelente programa de Certified Software Development Professional [32, 33] pero, aunque es excelente, no es completamente una certificación en Ingeniería de Software, porque desafortunadamente no exige experiencia en ciencia o en ingeniería. Además, al momento de escribir este artículo, sólo había 575 personas en el mundo que tienen la certificación CSDP. A partir de 1999, Texas comenzó a certificar a los ingenieros de software [34] y el Open Group ha establecido un programa de certificación en arquitectura de tecnología de la información [35].

## 5. CONCLUSIONES

La formación superior en los Estados Unidos necesita ser más sensible a las necesidades en Ingeniería de Software que tienen la industria y los laboratorios del gobierno. El país no puede ser complaciente con sus derivaciones tecnológicas en cualquier ámbito, especialmente en el software y en lo aeroespacial, que son dos de las industrias más importantes de su economía y que le suministran anualmente más de 180 y 900 billones de dólares, respectivamente. La tecnología ha ido cambiando con frecuencia y a un ritmo exponencial, mientras que los programas formativos cambian muy lentamente. La Ingeniería de Software necesita ser incorporada en los programas de ciencia e ingeniería existentes, especialmente en lo aeroespacial. También es necesario crear más programas formativos dedicados a la Ingeniería de Software en todos los niveles: cursos cortos, pregrados, especializaciones, maestrías y doctorados. Por último, también tiene que convertirse en un esfuerzo nacional el desarrollar la certificación profesional de los ingenieros de software.

## 6. REFERENCIAS

- [1] Jorgenson, D. W. & Wessner, C. W. 2006. [Measuring and Sustaining the New Economy, Software, Growth, and the Future of the U.S. Economy](#). National Academies Press.
- [2] NIST. 2010. [Software Testing Metrics Portal](#). Online. [Feb. 2012].
- [3] IEEE Computer Society & ACM. 2004. [Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering](#). National Science Foundation.
- [4] Vaughn, R. 2000. [Software Engineering Degree Programs](#). CROSSTALK, The Journal of Defense Software Engineering, 13(3), pp. 7-9.
- [5] Computer Science and Telecommunications Board. 2000. [Expanding Information Technology Research to Meet Society's Needs](#). The National Academies Press.
- [6] Sommerville, I. 2006. [Software Engineering](#). Addison-Wesley.
- [7] Glass, R. L. 1997. [Software Runaways: Monumental Software Disasters](#). Prentice-Hall.
- [8] Eggen, D. & Witte, G. 2006. [The FBI's Upgrade That Wasn't](#). The Washington Post, 18 Aug. Online: [Nov. 2011].
- [9] U.S. House of Representatives. 2001. [Proc. of the Aviation Subcommittee Meeting](#). Washington, D.C.
- [10] Leveson, N. G. 2004. [The Role of Software in Spacecraft Accidents](#). Journal of Spacecraft and Rockets, 41(4), pp. 1-27.
- [11] Winter, D. C. 2006. [Crucial Aviation Software and Systems Issues and Challenges](#). Presentation at the National Workshop on Aviation Software Systems: Design for Certifiably Dependable Systems. Alexandria, VA. Oct. 5-6.
- [12] The Economist. 2006. [The Aging Workforce: Turning Boomers into Boomerangs](#). Special report, 16 Feb. Online. [Dec. 2011].
- [13] Albaugh, J. F. 2005. [Embracing Risk: A Vision for Aerospace in the 21<sup>st</sup> Century](#). Frank Whittle Lecture, Royal Aeronautical Society, 19 Jan.
- [14] Commission on the Future of the U.S. Aerospace Industry. 2002. [Final Report of the Commission on the Future of the U.S. Aerospace Industry](#). National Academies Press.
- [15] Hafner, K. 1999. [Honey, I Programmed the Blanket](#). The New York Times, 27 May. Online. [Nov. 2011].
- [16] Pehrson, R. J. 1996. [Software Development for the Boeing 777](#). CrossTalk, Jan.
- [17] Moody, B. L. 2000. [F-22 Software Risk Reduction](#). CrossTalk, May.
- [18] Cook, D. A. & Dupaix, L. 1999. [A Gentle Introduction to Software Engineering](#). Software Technology Support Center.
- [19] Glass, R. L. 2006. [Facts and Fallacies of Software Engineering](#). Addison-Wesley.
- [20] [Ada Home](#). Online. [Jan. 2012].
- [21] Knight, J. C.; Graydon, P. J. & Strunk, E. A. 2006. [Basing aviation software certification on assurance cases](#). Presentation at the National Workshop on Aviation Software Systems: Design for Certifiably Dependable Systems. Alexandria, VA: Oct. 5-6.
- [22] American Institute of Physics. [The Statistical Research Center](#). Online. [Jan. 2011].
- [23] Knight, J. C. & Leveson, N. G. 2006. [Software and Higher Education](#). Communications of the ACM, 49(1), pp. 160.
- [24] Sanders, P. 1999. [Improving Software Engineering Practice](#). CrossTalk, Jan.
- [25] Committee on Aging Avionics in Military Aircraft. 2001. [Aging Avionics in Military Aircraft](#). National Academies Press.
- [26] Long, L. N. 2004. [Computing, Information, and Communication: The Fifth Pillar of Aerospace Engineering](#). Journal of Aerospace Computing, Information, and Communication, 1(1), pp. 1-4.
- [27] National Academy of Engineering. [The Future of Aerospace](#). National Academies Press.
- [28] Pennsylvania State University. [Curriculum Guide 2006-2007](#). Online. [Dec. 2011].
- [29] Committee on Engineering Education & National Academy of Engineering. 2005. [Educating the Engineer of 2020: Adapting Engineering Education to the New Century](#). National Academies Press.
- [30] [The Pennsylvania State University](#). [Jan. 2012].
- [31] ABET. [Dec. 2011].
- [32] IEEE Computer Society. [IEEE Computer Society Certified Software Development Professional Program Web Center](#). [Jan, 2012].
- [33] IEEE Computer Society. 2004. [Guide to the Software Engineering Body of Knowledge](#). IEEE Computer Society.
- [34] Voas, J. 1998. [The Software Quality Certification Triangle](#). CrossTalk Nov.
- [35] The Open Group. [IT Architect Certification Program](#). [Feb. 2012].