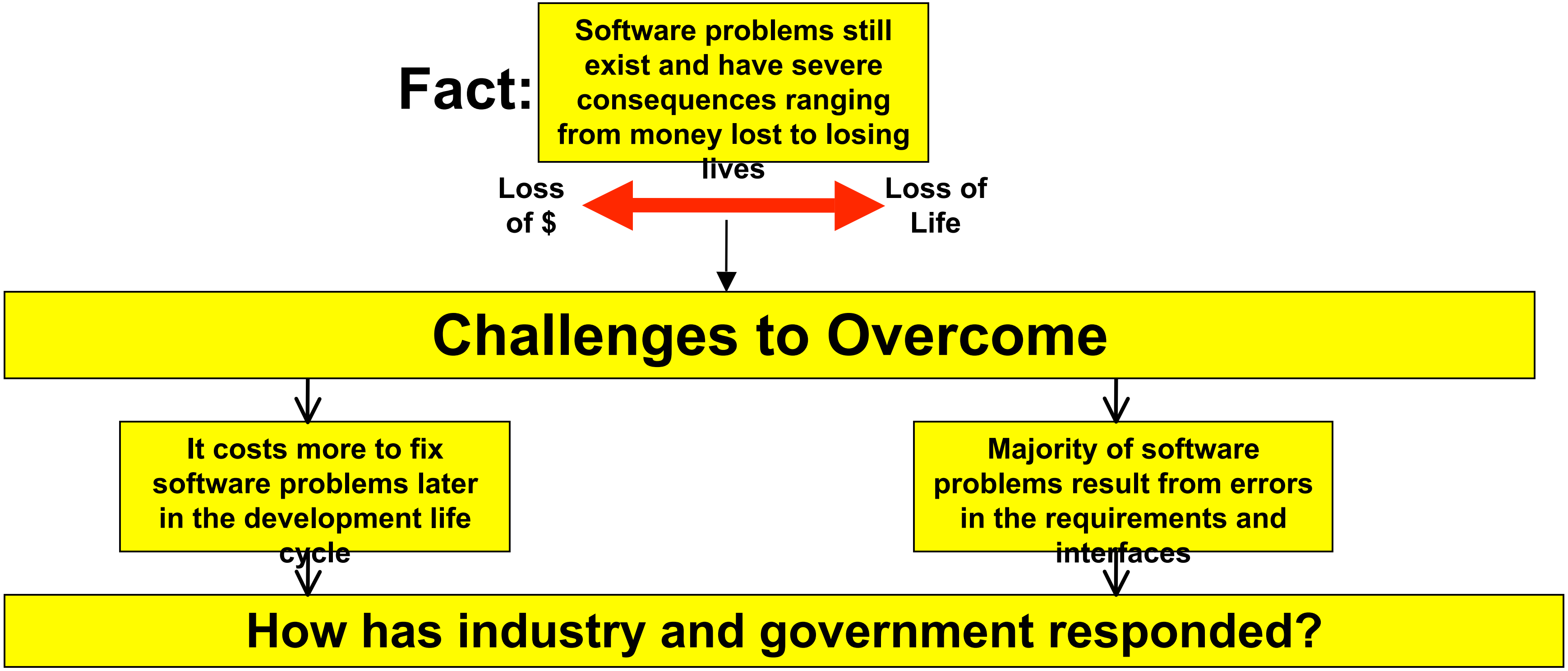
FAQ-03
What solutions have been applied to address the general problems?

How has Industry and Government Responded to the Challenges of Developing System Software?



Fact:

Software problems still exist and have severe consequences ranging from money lost to losing lives

Loss of $

Loss of Life

Challenges to Overcome

It costs more to fix software problems later in the development life cycle

Majority of software problems result from errors in the requirements and interfaces

How has industry and government responded?

- Improvements to the software development “Process”
  - They have made remarkable progression with respect to the advancement and standardization of Software Quality Standards and Development Practices
  - Systems are still being delivered with less functionality than expected
- Improvements to the software development “Products”
  - Industry and government introduced verification and validation activities to focus more on the quality of the software products (i.e. assuring the expected software behaviors are valid and delivered)

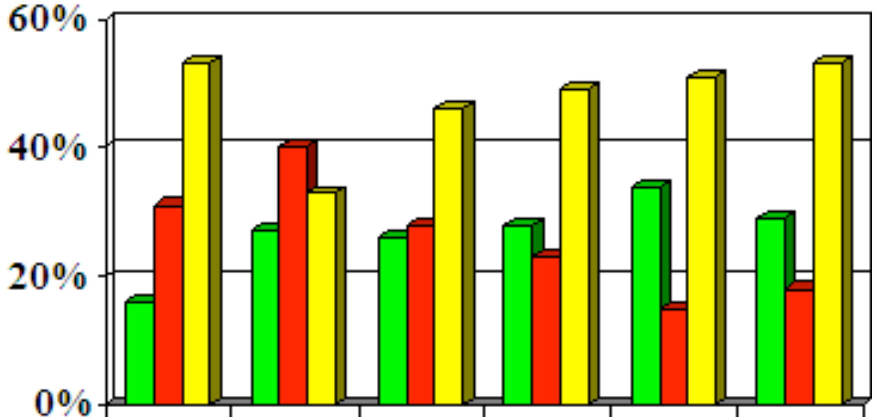
Improvements to the Process

- Responding to the challenges that exist with system software, industry & government placed an intense focus on software quality and reliability over the past 15 - 20 years.
- Software safety became more standardized in the 1990s[23]:
  - When a system is determined to be safety-critical (e.g., through a preliminary hazard analysis), the use of software within that system must be analyzed
    - Analysis begins with establishing the role software has within the entire system.
    - The analysis cannot be constrained by looking at the software components in isolation, simply because an accurate representation for the impact that software has on safety will not be captured (including the end-user of the system).
      - Software safety stresses that software cannot be divorced from the system where it resides.
    - Software safety analyses conclude by evaluating how well the software safety requirements are defined, designed, and implemented in the system.
- Academia recognized the importance of the problem:
  - Software Quality, Software Engineering ,and Software Systems Engineering are disciplines with established curriculums up to and including the PhD levels.
- Many different software-related organizations and processes have emerged:
  - Assurance, Software Safety, Software Reliability, Software Verification and Validation, Software IV&V, Independent Testing, CMMI, ISO-9001(2000), Software Metrics, Beta Testing, Continuous Improvement, Risk Management, etc.

Improvements to the Process (cont)

- National Institute for Standards and Technology (NIST) stresses an approach enabling early identification and removal of software defects
  - NIST concluded that $22.2 billion lost could have been eliminated by an improved approach.[2]
- The Borland company has enhanced their practices for assuring that the requirements are correct
  - Borland concluded that rework typically accounted for 40% of a development budget and most of the rework focused on correcting software requirements defects.[25]
- Many software related industry standards have been developed:
  - IEEE Std 610.12-1990 (Reaff 2002), IEEE Standard Glossary of Software Engineering Terminology.
  - IEEE Std 829-1998, IEEE Standard for Software Test Documentation.
  - IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software.
  - IEEE Std 1012A™-1998, Supplement to IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997.
  - IEEE Std 1028-1997, IEEE Standard for Software Reviews.
  - IEEE Std 1044-1993, IEEE Standard for Software Anomalies.
  - IEEE Std 1061-1998, IEEE Standard for a Software Quality Metrics Methodology.
  - IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Processes.
  - IEEE 1228, Standard for Software Safety Plans
  - IEEE Std 1517-1999, IEEE Standard for Information Technology - Software Life Cycle Processes - Reuse Processes.
  - IEEE/EIA Std 12207.0-1996, IEEE/EIA Standard—Industry Implementation of International Standard
  - ISO/IEC 12207:1995 (ISO/IEC 12207) Standard for Information Technology—Software Life Cycle Processes.
  - ISO/IEC 12207:1995, Information Technology—Software Life Cycle Processes; as amended by Amendment 1:2002.7
  - ISO 8402, Quality Management and Quality Assurance – Vocabulary
  - RTCA DO-178B, Software Considerations in Airborne Systems and Equipment Certification
  - DOD Joint Software System Safety Handbook

The Affect on Software Development



Reference: Standish Group

	1994	1996	1998	2000	2002	2004
🟩 Succeeded	16%	27%	26%	28%	34%	29%
🟥 Failed	31%	40%	28%	23%	15%	18%
🟨 Challenged	53%	33%	46%	49%	51%	53%

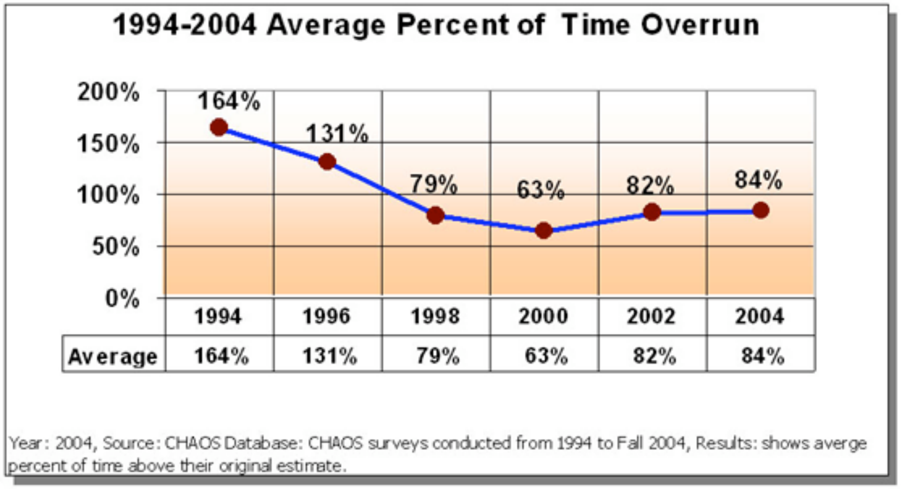
Success rates seemed to have stabilized, indicating that any more changes to the “Process” would not have any significant affect on the success rate.

Industry has made remarkable progression with respect to the advancement and standardization of Software Quality Standards and Development Practices over the past 15 years – most of this has focused on the “Process”.

ISO and CMMI have strengthened these attributes

Latest Standish Group CHAOS Report (2004) Shows Project Success Rates Have Improved by 50%
Project success rates have increased to just over a third or 34% of all projects. This is a 100% plus improvement over the 16% rate in 1994. Project failures have declined to 15% of all projects, which is more than half the 31% in 1994. Challenged projects account for the remaining 51%.

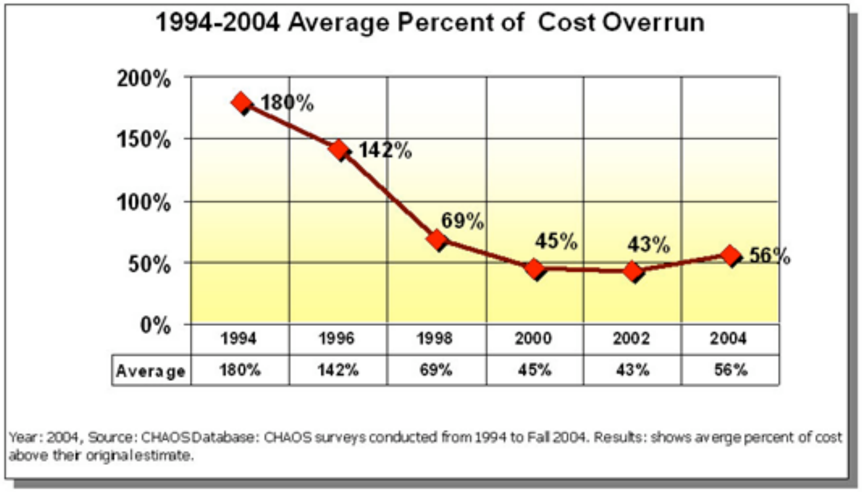
Affect on Software Development (cont)



1994-2004 Average Percent of Time Overrun

	1994	1996	1998	2000	2002	2004
Average	164%	131%	79%	63%	82%	84%

Year: 2004, Source: CHAOS Database: CHAOS surveys conducted from 1994 to Fall 2004, Results: shows averge percent of time above their original estimate.



1994-2004 Average Percent of Cost Overrun

	1994	1996	1998	2000	2002	2004
Average	180%	142%	69%	45%	43%	56%

Year: 2004, Source: CHAOS Database: CHAOS surveys conducted from 1994 to Fall 2004. Results: shows averge percent of cost above their orignal estimate.

While Industry's solution has improved the <u>Process</u> for developing software, the data still suggests the <u>Product</u> requires attention!

In 2003, the Standish Group reported that software developers fielded <u>only 52% of the required features and functions</u> as compared to 67% in the year 2000.

As such, IV&V places a heavy emphasis on getting in early, staying in phase, and focusing on the software "Product" (i.e. assuring the requirements are correct).

Improvement to the "Products"

- While improvements to the software development process have been successful, industry and government have been additionally focusing on improving the software products through V&V and IV&V.

- IEEE defines Validation as:
  - Provides evidence whether the software and its associated products and processes:
    - Satisfy intended use and user needs
    - Satisfy system requirements allocated to software at the end of each life cycle activity
    - Solve the right problem (e.g., correctly model physical laws, implement business rules, use the proper system assumptions)
- IEEE defines Verification as:
  - Provides objective evidence whether the software and its associated products and processes:
    - Conform to requirements (e.g., for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance)
    - Satisfy standards, practices, and conventions during life cycle processes
    - Successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities (e.g., building the software correctly)

Improvement to the “Products” (cont)

- IEEE defines independence in IV&V in terms of three parameters:
  - Technical Independence requires the V&V effort to utilize personnel who are not involved in the development of the software.
    - The IV&V effort should formulate its own understanding of the problem and how the proposed system is solving the problem. Technical independence (“fresh viewpoint”) is an important method to detect subtle errors overlooked by those too close to the solution.
    - For software tools, technical independence means that the IV&V effort uses or develops its own set of test and analysis tools separate from the developer’s tools.
  - Managerial Independence requires that the responsibility for the IV&V effort be vested in an organization separate from the development and program management organizations.
    - The IV&V effort independently selects the segments of the software and system to analyze and test, chooses the IV&V techniques, defines the schedule of IV&V activities, and selects the specific technical issues and problems to act upon.
    - The IV&V effort provides its findings in a timely fashion simultaneously to both the development and program management organizations.
  - Financial Independence requires that control of the IV&V budget be vested in an organization independent of the development organization.

Improvement to the "Products" (cont)

- The U.S. Nuclear Regulatory Commission (NRC) uses a combination of their Project Management Methodology and IV&V to succeed in deploying the National Source Tracking System (NSTS).[12]
  - NRC recognizes IV&V as a best practice and utilizes IV&V extensively on the National Source Tracking System (NSTS)[26]
    - IV&V is a very effective tool for dealing with uncertainty
    - The degree of IV&V can be tailored to the size, complexity, and importance of the system
- IV&V at US Department of Justice (DOJ)[28]
  - DOJ uses IV&V to improve their software products
  - IV&V is an independent entity that assesses the system as it is developed in order to evaluate if the software will perform as intended

Improvement to the “Products” (cont)

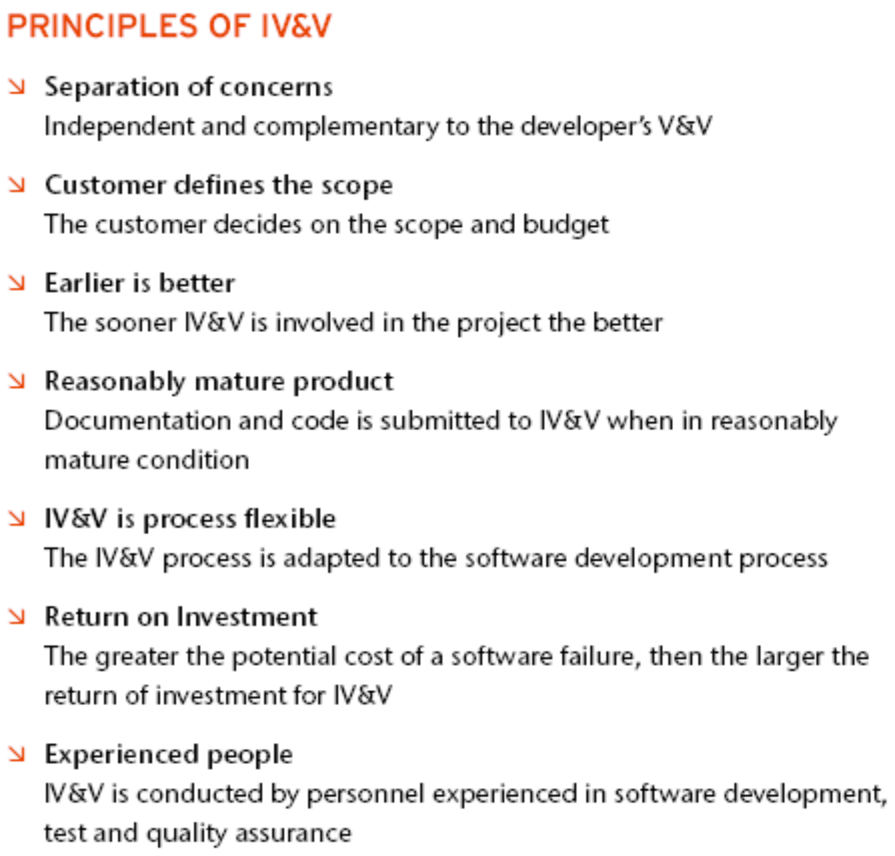
- **Department of Veterans Affairs uses IV&V to improve their software**[27]
  - **IV&V is defined as systems verification and validation *performed independently* of the program office and development team for an automated information system.**
  - **IV&V’s purpose is to help the program and development organizations to *build quality into system software* during the system development life cycle.**
  - **IV&V activities determine whether development products conform to the requirements and *whether the system software satisfies the intended use* and specified needs of the user.**
  - **IV&V is an *extension of* both program management and *system engineering functions*.**
  - ***System integrity levels* and *risk assessments* are used to determine the appropriate level of IV&V activities**
  - **Through assessment, analysis, evaluation, review, inspection…. of software products, IV&V identifies objective data and conclusions about software quality, performance, application security….. for the program and development organizations.**
    - **These results allow the program to *modify the software products* in a *timely fashion* and *reduce project costs* and *schedule delays*.**
  - **This *proactive approach of employing IV&V in parallel with the system development life cycle*, enables programs to *address anomalies and defects***

Improvement to the "Products" (cont)

- The Department of Defense (DOD) utilizes V&V as part of their systems engineering role[30]
  - Validation answers the question "Did you build the right thing" by testing the performance of systems within their intended operational environment, with anticipated operators and users.
    - In the early stages of the system life cycle, validation may involve prototypes, simulations, or mock-ups of the system and a model or simulation of the system's intended operational environment.
  - Verification confirms that the system element meets the design-to or build-to specifications.
    - Verification answers the question "Did you build it right?" by testing the system elements against their defined requirements.
    - The purpose of their Verification is to:
      - Conduct verification of the realized (implemented or integrated) system element (including interfaces) from the lowest level system element up to the total system to ensure that the realized product conforms to the build-to specifications;
      - Generate evidence necessary to confirm that system elements at each level of the system hierarchy meet their build-to specifications; and
      - Verify the materials employed in system solutions can be used in a safe and environmentally compliant manner.

Improvement to the "Products" (cont)

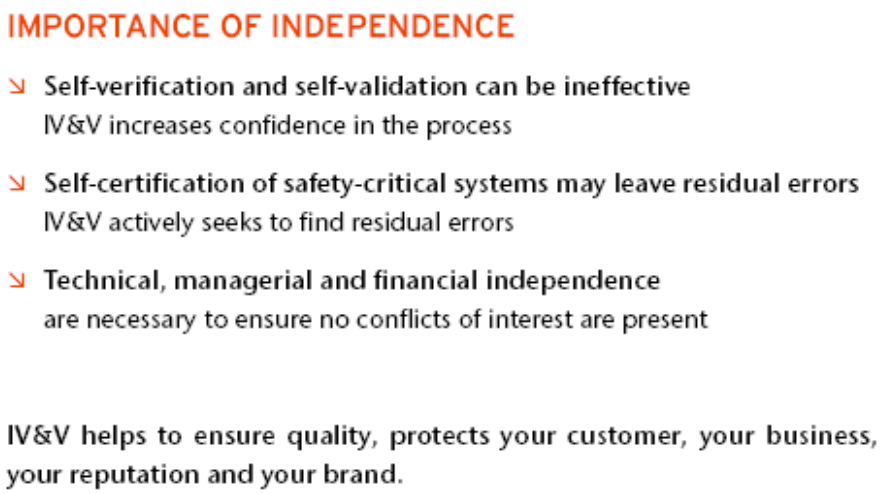
- The Automotive Industry follows these principles in employing IV&V[17]



PRINCIPLES OF IV&V

- **Separation of concerns**
  Independent and complementary to the developer's V&V
- **Customer defines the scope**
  The customer decides on the scope and budget
- **Earlier is better**
  The sooner IV&V is involved in the project the better
- **Reasonably mature product**
  Documentation and code is submitted to IV&V when in reasonably mature condition
- **IV&V is process flexible**
  The IV&V process is adapted to the software development process
- **Return on Investment**
  The greater the potential cost of a software failure, then the larger the return of investment for IV&V
- **Experienced people**
  IV&V is conducted by personnel experienced in software development, test and quality assurance

- The Automotive Industry stresses the importance of independence[17]



IMPORTANCE OF INDEPENDENCE

- **Self-verification and self-validation can be ineffective**
  IV&V increases confidence in the process
- **Self-certification of safety-critical systems may leave residual errors**
  IV&V actively seeks to find residual errors
- **Technical, managerial and financial independence**
  are necessary to ensure no conflicts of interest are present

IV&V helps to ensure quality, protects your customer, your business, your reputation and your brand.

Improvement to the "Products" (cont)

- Cross Talk, The Journal of Defense Software Engineering, described a methodology to remedy the common problems being realized across industry[31]
  - Requirements and specifications are incomplete
  - Requirements and specifications change too often
  - There is a lack of user input (to requirements)
- Their methodology addresses each of these issues
  - It begins at the first phase of software development where the correction of error is the least costly
  - It begins at the requirements phase where the largest portion of bugs have their root cause
  - It addresses improving the quality of requirements: Inadequate requirements often are the reason for failing projects.

Improvement to the “Products” (cont)

The RBT (Requirements-Based Testing) methodology is a 12-step process. Each of these steps is described below.

1. Validate requirements against objectives.
2. Apply use cases against requirements.
3. Perform an initial ambiguity review.
4. Perform domain expert reviews.
5. Create cause-effect graph.
6. Logical consistency checks performed and test cases designed.
7. Review of test cases by requirements authors.
8. Validate test cases with the users/domain experts.
9. Review of test cases by developers.
10. Use test cases in design review.
11. Use test cases in code review.
12. Verify code against the test cases derived from requirements.

Note: Once all of the test cases execute successfully against the code, then it can be said that 100 percent of the functionality has been verified and the code is ready to be delivered into production. From a CMMI perspective, you have verified that you are *building the system right*.

What is the value/benefit of IV&V?

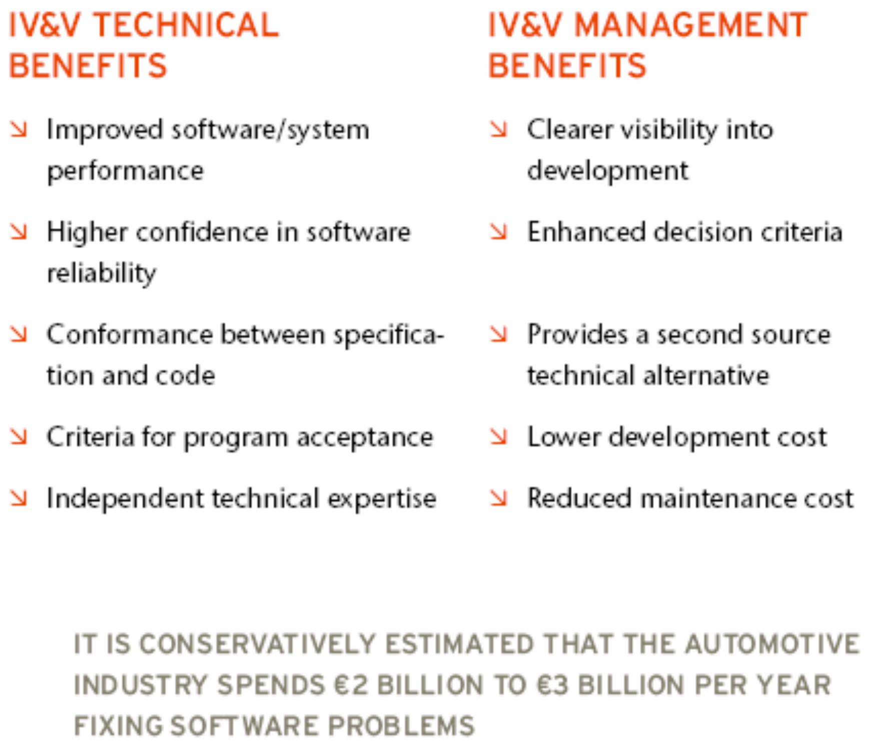
- GAO report[29] on DoD Navy
  - "The use of an IV&V function is ***recognized as a best business practice*** and can help provide reasonable ***assurance*** that the ***system satisfies its intended use and user needs"***
  - "IV&V function should report on every facet of a system project such as…… ***Testing program adequacy.*** Testing activities would be evaluated to ensure they are properly defined and developed in accordance with industry standard and best practices".
  - "…performing IV&V activities ***independently of the development and management functions*** helps to ensure that the results are unbiased and based on objective evidence"

What is the value/benefit of IV&V? (cont)

According to Rovsing a leader in Automotive and ESA IV&V:

The automotive industry has realized the following benefits of IV&V [17]

- **Higher confidence** in the software safety and dependability
- **Reduced development risk, time and cost** with earlier detection of errors
- **Conformance to requirements** fewer discrepancies between requirements and product
- **Better decision criteria** through clearer visibility into the development process
- **Higher reliability and robustness of the product** resulting from independent stress testing
- **Increased usability and maintainability** with higher quality and consistency of documentation



IV&V TECHNICAL BENEFITS	IV&V MANAGEMENT BENEFITS
↘ Improved software/system performance	↘ Clearer visibility into development
↘ Higher confidence in software reliability	↘ Enhanced decision criteria
↘ Conformance between specification and code	↘ Provides a second source technical alternative
↘ Criteria for program acceptance	↘ Lower development cost
↘ Independent technical expertise	↘ Reduced maintenance cost

IT IS CONSERVATIVELY ESTIMATED THAT THE AUTOMOTIVE INDUSTRY SPENDS €2 BILLION TO €3 BILLION PER YEAR FIXING SOFTWARE PROBLEMS

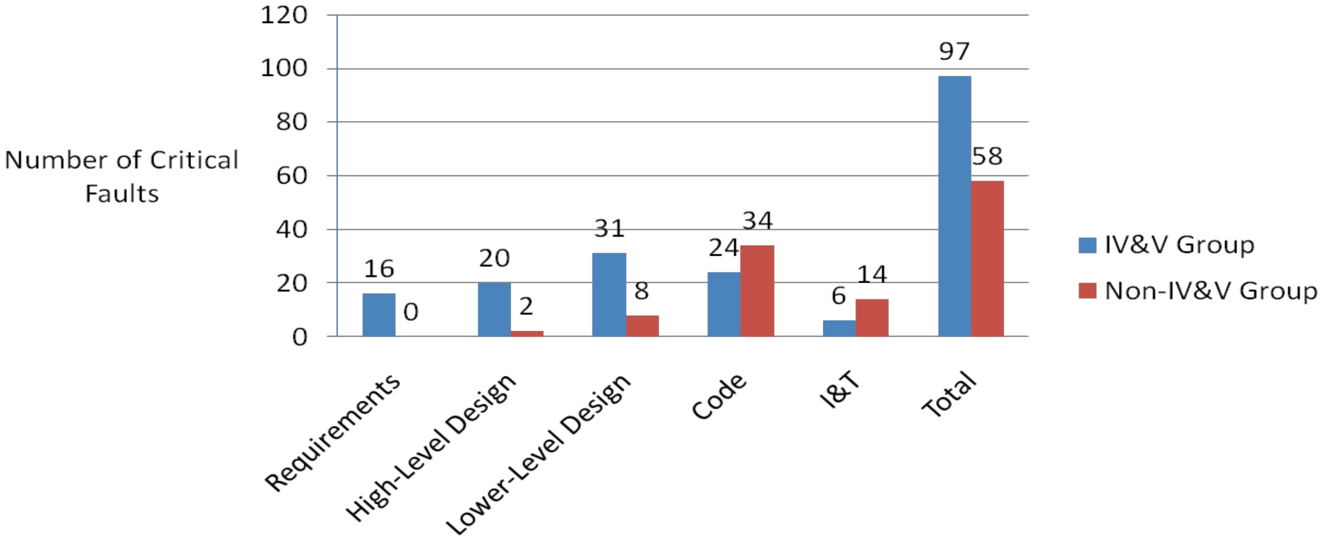
What is the value/benefit of IV&V? (cont)

- External studies using industry and publically available data have been conducted and attest to the value of IV&V
  - Studies estimate that development efforts that initiate IV&V at the beginning of the requirements phase experience a savings of 92%-180% the costs of IV&V[32]
  - Benefits cited are enumerated in two case studies that specifically compare the projects developed under the auspices of IV&V to those with no IV&V. Benefits include:
    - IV&V promotes objectivity in that it helps maintain an unbiased technical viewpoint and supports an objective engineering analysis
    - IV&V finds software and system errors earlier
    - Earlier error detection translates into reduced effort and cost in removing those errors
    - Enhanced operational correctness is a direct benefit of IV&V
    - IV&V results in reduced variability in the development process

"verification and validation activities produce their best results when performed by a V&V agent who operates independently of the developer or specification agent" B. Boehm 1984

What is the value/benefit of IV&V? (cont)

- **NASA LaRC funded a joint effort with Virginia Tech to examine the effectiveness of applying an independent V&V methodology**
  - **Study used two groups to assess the extent IV&V activities help detect faults earlier in the development life cycle, reduce the time to remove those faults, and produce a more robust product.**



Findings:

1. **IV&V Group found 67% more critical faults than the Non-IV&V Group.**
2. **IV&V Group found 67 critical faults before coding while Non-IV&V Group found only 10 critical faults before coding**

Bibliography

2. Press Release (for Planning Report 02-3 referenced below) National Institute of Standards and Technology (NIST) June 28, 2002
   http://www.nist.gov/public_affairs/releases/n02-10.htm

3. Planning Report 02-3, The Economic Impacts of Inadequate Infrastructure for Software Testing National Institute of Standards and Technology (NIST); May 2002
   http://www.nist.gov/director/prog-ofc/report02-3.pdf

7. General Principles of Software Validation; Final Guidance for Industry and FDA Staff
   U.S. Food and Drug Administration; January 11, 2002
   http://www.fda.gov/cdrh/comp/guidance/938.html

12. U.S. Nuclear Regulatory Commission - Fiscal Year 2005 Annual Report To The Office Of Management and Budget On E-Government Act of 2002; October 21, 2005
    http://www.nrc.gov/reading-rm/doc-collections/e-gov/ml052700022.pdf

15. Greedy to Get the Bugs Out
    Linley Erin Hall; Spring 2005
    http://researchmag.asu.edu/stories/bugsout.html

17. Independent Verification and Validation of Automotive Software, Rovsing
    http://webserver.rovsing.dk/uploads/File/brochures/Automotive_IVV_small.pdf

18. Rovsing Company History
    http://webserver.rovsing.dk/index.php?page=company-history

19. Rosving's Independent Verification and Validation of Automotive Software
    http://webserver.rovsing.dk/index.php?page=independent-verification-and-validation-4

20. U.S. Nuclear Regulatory Commission (NRC) - National Source Tracking System (NSTS)
    http://www.nrc.gov/security/byproduct/nsts.html#history

21. Testimony to Software Failures and Famous Bugs….
    Fanshawe College, School of Information Technology
    http://infotech.fanshawec.ca/gsantor/Computing/FamousBugs.htm

22. Software Assurance, Department of Homeland Security
    http://www.uscert.gov/reading_room/infosheet_SoftwareAssurance.pdf

23. Leveson, Nancy, Safeware - System Safety and Computers, Addison Wesley, 1995, Reading, MA, 680 pp.
    http://sunnyday.mit.edu/papers/therac.pdf

24. Software Hall of Shame
    http://spectrum.ieee.org/sep05/1685/failt1

25. Software Requirements Definition
    http://www.borland.com/us/solutions/requirements-definition-management/requirements-definition.html

26. NRC Commission Paper, Oct 2001
    http://www.nrc.gov/reading-rm/doc-collections/commission/secys/2001/secy2001-0187/2001-0187scy.html

Bibliography (cont)

27. Reference: VA Directive 4900 INTERNAL CONTROLS FOR FINANCIAL AND FINANCIAL INTERFACING AUTOMATED INFORMATION SYSTEMS (June 2004) http://www1.va.gov/vapubs/viewPublication.asp?Pub_ID=164&FType=2

28. USDOJ/OIG, Status of the Federal Bureau of Investigation's Case Management System, Audit Report 07-40, August 2007 http://www.usdoj.gov/oig/reports/FBI/a0740/intro.htm

29. 1 GAO-05-858 US GAO Report to Congressional Requesters, DoD Business Systems Modernization: Navy ERP Adherence to Best Business Practices Critical to Avoid Past Failures, Sept 2005 http://www.gao.gov/new.items/d05858.pdf

30.https://akss.dau.mil/dag/DoD5000.asp?view=document&rf=GuideBook\IG_c4.2.4.7.asp

31.http://www.stsc.hill.af.mil/CrossTalk/2003/03/mogyorodi.html

32. Wallace, D.R. and R.U. Fujii 1989. Software Verification and Validation: An Overview, *IEEE Computer* 6(3): 10-17.