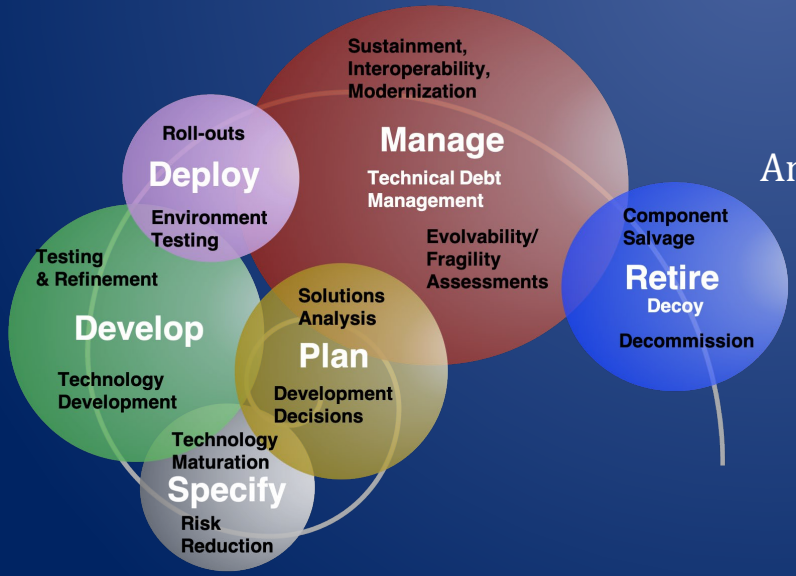
NAVAL POSTGRADUATE SCHOOL

Addressing Software-based, Platform Interoperability Risks in Defense Systems by Using Distressed Debt Financial Strategies: A Technical Debt Mitigation Concept



Ann Gallenson, Scot Miller, & Susan Higgins
Naval Postgraduate School, Monterey, CA
Acquisition Research Symposium, 2021

Legacy Code in Combat Systems

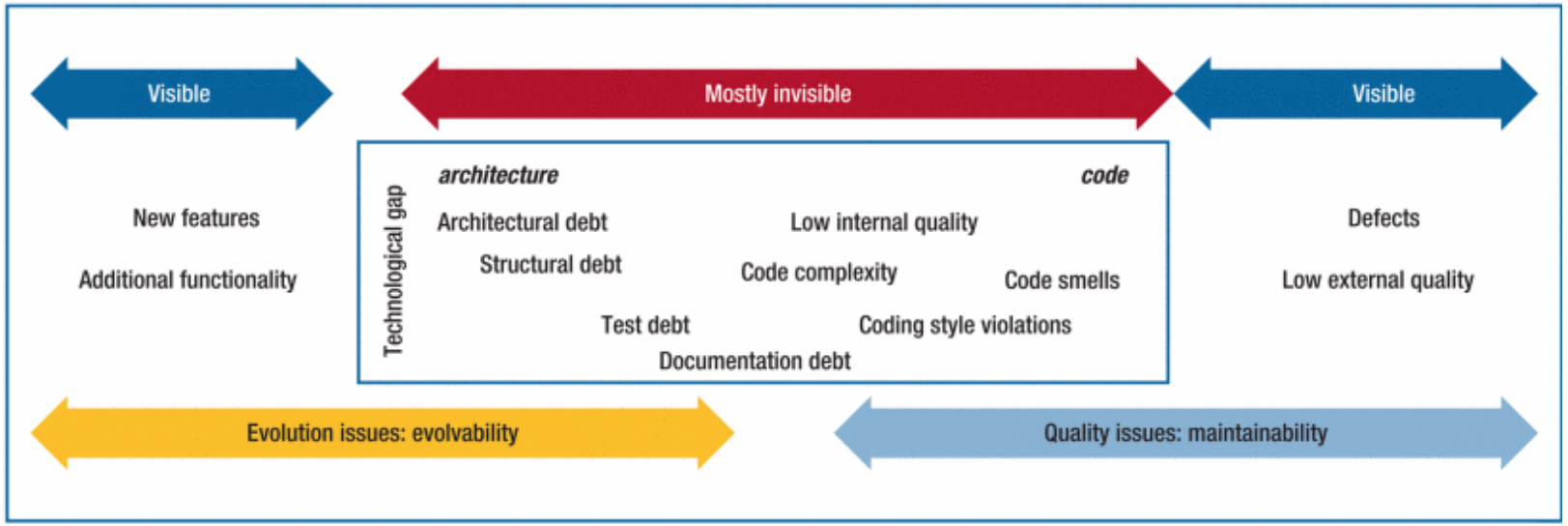
Over the past 60 years the Navy's critical combat, command, control, and communications systems have evolved into software-dependent technological ecosystems. All software systems are subject to the impacts of technical debt. Longevity, interoperability and integration requirements make prevention and early mitigation of technical debt a mission critical practice.

Research Questions:

1) What technical methods are being used to detect, assess, monitor, and mitigate technical debt and realign/re-introduce refactored code into existing systems?
2) What program management practices can best meet the time and technical demands of maintaining platform readiness while addressing complex, embedded software reliability issues?

Technical Debt Landscape

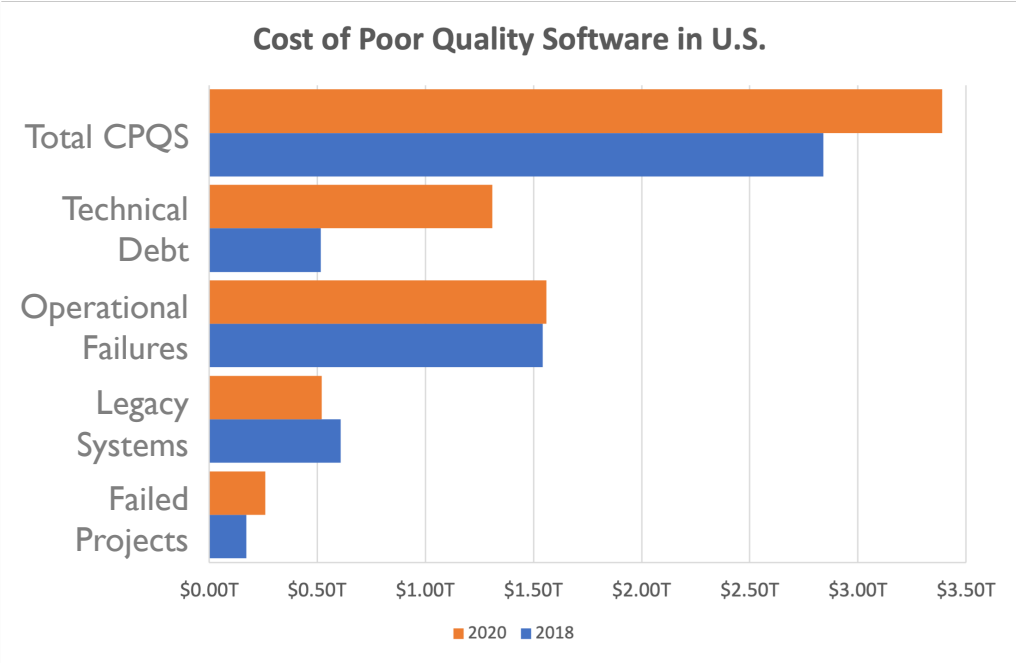
Technical debt (TD) is a term used for the inevitable software design and implementation tradeoffs taken to meet short term business and development goals. Technical debt accrues throughout the life of the software and compounds if left unmitigated. Aging code can hinder interoperability, maintenance, and evolvability. Technical Debt equates to mission risk and fragility.



Krutchen, 2019, p. 5

Findings: Technical Debt and its Impact

- CISQ reported the cost of poor-quality software in the US in 2020 was $2.08T plus $1.31T in deferred TD.
- By 2021 Cybercrime will cost companies $2.08T annually due to unsuccessful projects, legacy systems, and software failures across the industry.
- Federal IT Professionals report that TD directly impacts the speed and cost of providing new functionality.
- Interoperability, COTS-based modifications, machine learning injections, and system add-ons can increase and complicate existing TD.

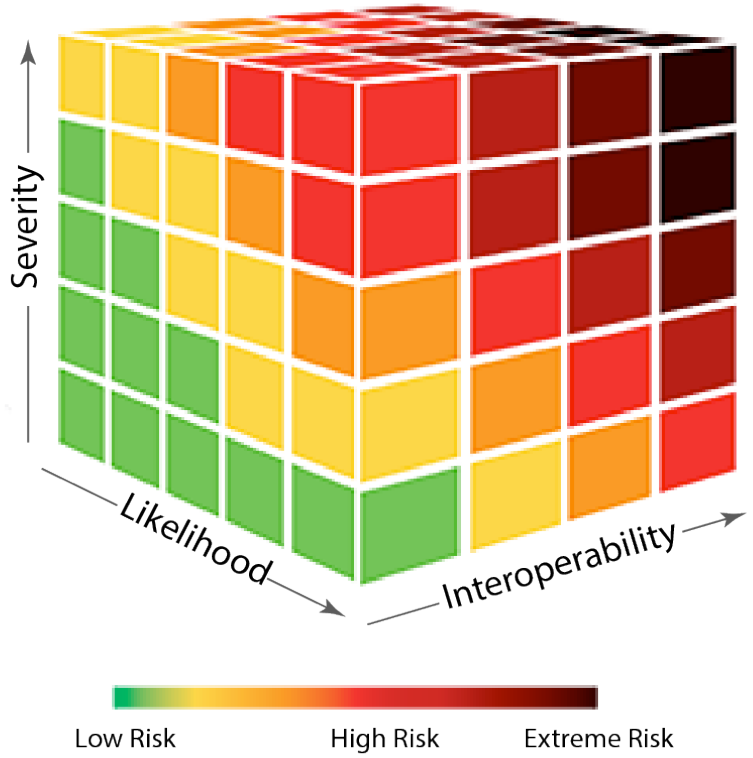


Compiled from Krasner, CISQ, 2019 and 2021

CISQ uses publicly available data for its estimates; the resulting compilations are underestimates due to the reluctance of firms to negatively impact the value of publicly-held assets.

Extending Risk Assessment with Interoperability

- The specialized nature of weapons and weapon systems made the programs early adopters of cyber-physical integration.
- Continual hardware and software modernization resulted in complicated middleware processes for interoperability.
- Adding interoperability and integration into existing risk calculations can expose vulnerabilities across networked systems and elevate the urgency to act.



Recommendation:

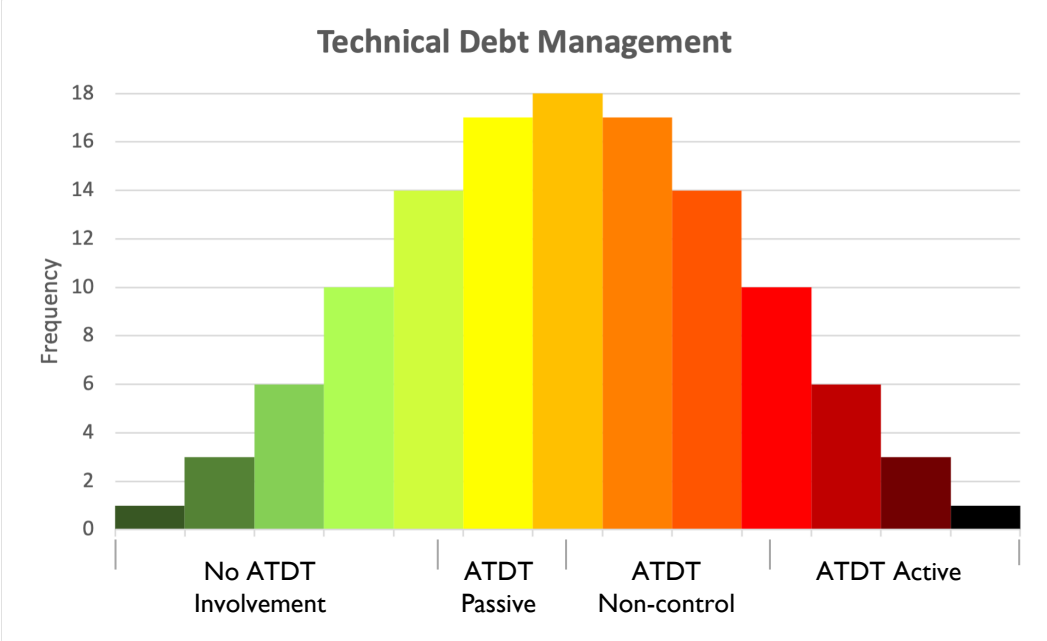
Create an Acquisition Technical Debt Team (ATDT)

- To mitigate TD-related functional or interoperability weapon systems' risks we, recommend creating a specialized "blue team" of maintainers and their program managers to focus on TD issues.
- ATDT assists program managers and maintainers of software intense systems with the complexities of managing and mitigating technical debt across their system portfolios.
- Build funding and timeline guidelines and advocate with program sponsors for the resources to execute solutions.

Build from Destressed Debt Management Methods

Combining the categories from the interoperability risk cube and Distressed Debt Management methods, we can classify the type of involvement of the ATDT.

- **No ATDT involvement** – if the risk is within manageable levels.
- **ATDT Passive involvement** – As risk increases to ”yellow,” the problem should be activated to an ATDT watch list and the ATDT members selected.
- **ATDT Active Non-control** – As the risk increases to “orange,” the ATDT should be assembled, and funding infusions and timeline modifications arranged.



- **ATDT Active Control** – As the risks increases to “orange-red through black,” the ATDT will begin active, collaborative management of the technical debt. Mitigation plans, costs, and specifications should be created. Timelines for refactoring or replacement, testing and reintroduction should be established.

Conclusion and Next Steps

Technical debt infects all software intense systems across the services. A system-wide threat requires a system-wide response.

- Select a combat system that is already dealing with technical debt issues to learn their existing technical debt evaluation processes and create a risk cube from system data.
- Create a cross functional team familiar with the combat system and the technical issues that are contributing to vulnerability concerns.
- Evaluate the interoperability risk cube and the technical debt management strategies suggested here to learn the relevance of these models to the combat system's needs.
- Continue this research to refine these concepts with the Acquisition and Warfare Center communities.

References

Curtis, W. (2018, February 23). Solving the 'technical debt' problem. FCW. https://fcw.com/articles/2018/02/23/comment-technical-debt-curtis.aspx

Chaplain, C. (2018). Weapon Systems Cybersecurity: DoD just beginning to grapple with scale of vulnerabilities. Washington, DC, USA, GAO Report No. GAO-19-128.

Jain, S. (2011). Investing in distressed debt. *UBS Alternative Investments, June*, *15*.

Krasner, H. (2019). The cost of poor quality software in the us: A 2018 report. *Consortium for IT Software Quality, Tech. Rep*, *10*.

Krasner, H. (2021). The cost of poor quality software in the us: A 2020 report. *Consortium for IT Software Quality, Tech. Rep*, *10*.

Kruchten, P., Nord, R., & Ozkaya, I. (2012). Technical debt: From metaphor to theory and practice. IEEE Software, 29(6), 18–21.

Kruchten, P., Nord, R., & Ozkaya, I. (2019). Managing Technical Debt: Reducing Friction in Software Development. Addison-Wesley Professional.

Li, Z., Avgeriou, P., & Liang, P. (2015). A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, *101*, 193-220.

Object Management Group (OMG). (2017, December). Automated Technical Debt Measure Specification (Version 1.0) [Computer software specifications]. Consortium for IT Software Quality. https://www.omg.org/spec/ATDM

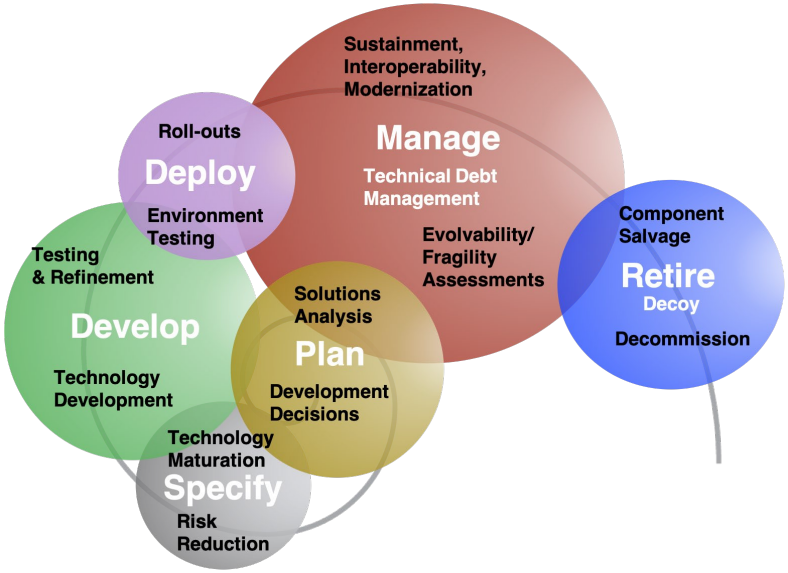
Rantala, L. (2020, October). Towards better technical debt detection with NLP and machine learning methods. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)* (pp. 242-245). IEEE.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Dennison, D. (2015). Hidden technical debt in machine learning systems. In Advances in neural information processing systems (pp. 2503-2511).

Tsoukalas, D., Kehagias, D., Siavvas, M., & Chatzigeorgiou, A. (2020). Technical debt forecasting: an empirical study on open-source repositories. Journal of Systems and Software, 170, 110777.

Yang, Y., Wade, J., Alelyani, T., & Stanton, P. (2018). RT 193: Framework for Analyzing Versioning and Technical Debt. STEVENS INST OF TECH HOBOKEN NJ HOBOKEN United States.

Addressing Software-based, Platform Interoperability Risks in Defense Systems by Using Distressed Debt Financial Strategies: A Technical Debt Mitigation Concept



Questions and Suggestions?

NAVAL
POSTGRADUATE
SCHOOL