Solution Debt in the Age of Digital Engineering

Matthew Taylor, Dr. Heidi Davidz, Dr. Doug Orellana

Agenda

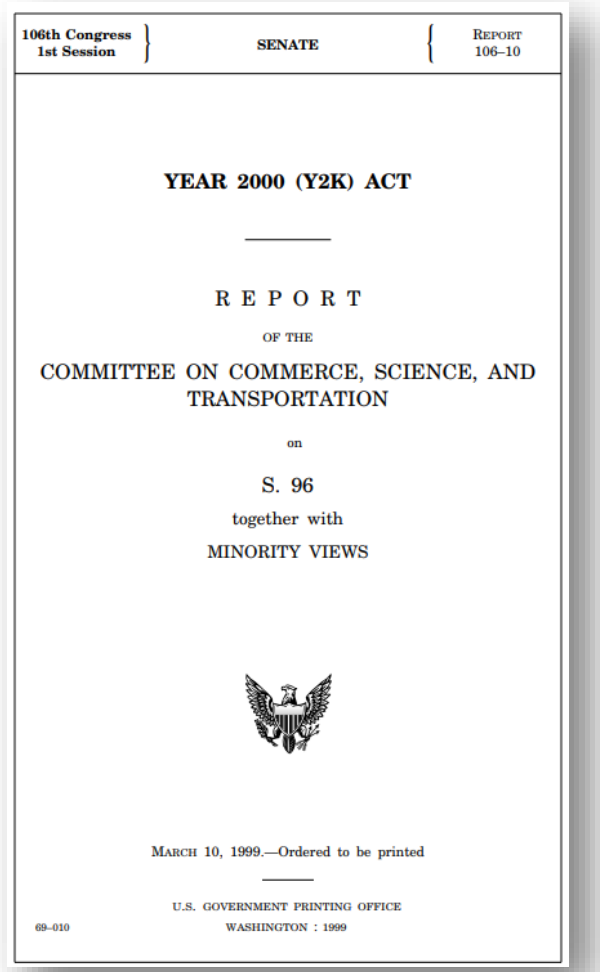
The Debt Metaphor → Solution Debt → Digital Engineering and Debt → Evaluating Your Debt

- The traditional definitions of debt are too narrow in the context of complex missions and systems
- Solution debt attempts a more comprehensive view across discipline and lifecycle
- Digital Engineering approaches avoid some debts, but magnify others
- The Solution debt concept can provide a useful context to categorize and evaluate debt

Y2K – It became a punchline, but had a real cost…[24,25,26,27]

Estimates of up to **$1 Trillion** spent in just the United States

- Estimates of $8.5B spent by the federal government
- $250M by Chase Manhattan Bank alone


106th Congress
1st Session

SENATE

REPORT
106–10

YEAR 2000 (Y2K) ACT

REPORT

OF THE

COMMITTEE ON COMMERCE, SCIENCE, AND TRANSPORTATION

on

S. 96

together with

MINORITY VIEWS

MARCH 10, 1999.—Ordered to be printed

U.S. GOVERNMENT PRINTING OFFICE

69–010 WASHINGTON : 1999

…all because memory and processing used to be so precious most software stored only two digits of the year – even after that wasn't strictly necessary.

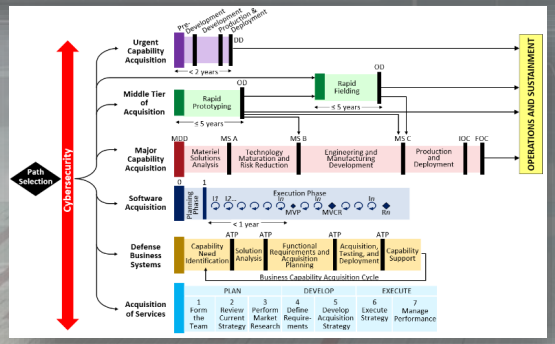
This was brought up as a debt to be paid as far back as 1971.

Common Debt Causes & Industry Forces (1 of 2) 1,2,3,4,5,6,7

Time Pressure

Limited Skills

DoD Restructures Acquisition, Technology Office to Improve Military Lethality, Speed



DEFENSE INNOVATION UNIT

Gartner

Lack of Skills Threatens Digital Transformation

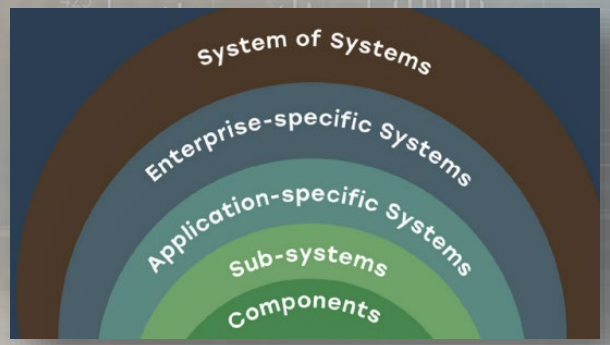
HBR

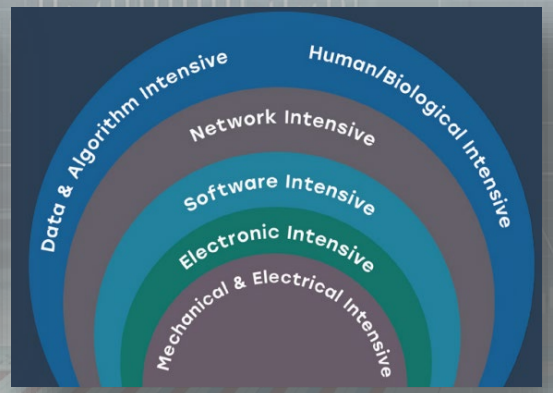
Digital Transformation Is About Talent, Not Technology

Common Debt Causes & Industry Forces (2 of 2) [1,8,9,10]

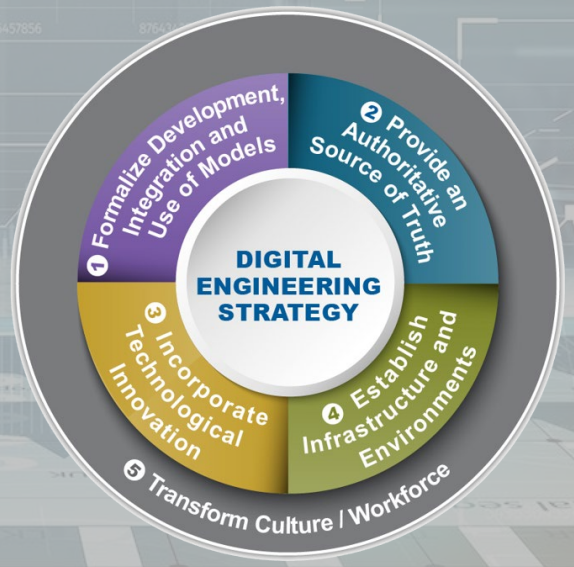
Complex Design

Limited Standards





SUMMARY OF THE
JOINT ALL-DOMAIN COMMAND & CONTROL
(JADC2)
STRATEGY



How standardized is DE today? In terms of…

- Technologies
- Processes
- Methods
- Tools
- Languages
- Environments
- Industry Standards

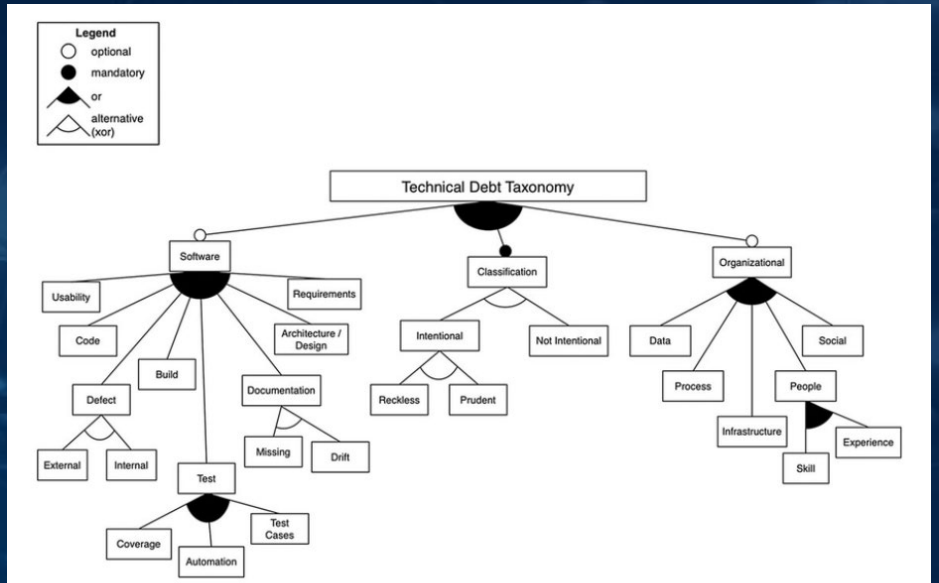
The Debt Metaphor… Is not just for Software [17,18,19,20]
… Is not Limited to Development Phases

Tech debt literature is almost exclusively focused on software.

"Technical debt is a concept in **programming**…" [18]

"Tech debt, also known as technical debt or **code debt**…" [19]

"technical debt, a term **specific to software** application **design and development**." [20]



Technical debt taxonomy

Systems Engineers & Enterprise Architects…
Worry about debt incurred at the solution level and spanning the lifecycle!

Introducing a Broader Term… Solution Debt

The traditional definition of technical debt is too narrow for the concerns of Systems Engineers and Enterprise Architects.

SE's and EA's care deeply about software, but also about many other facets of technology and business.

What are the debt drivers in delivering an overall solution? What will be more expensive in the solution or its ecosystem to fix or pay back later?

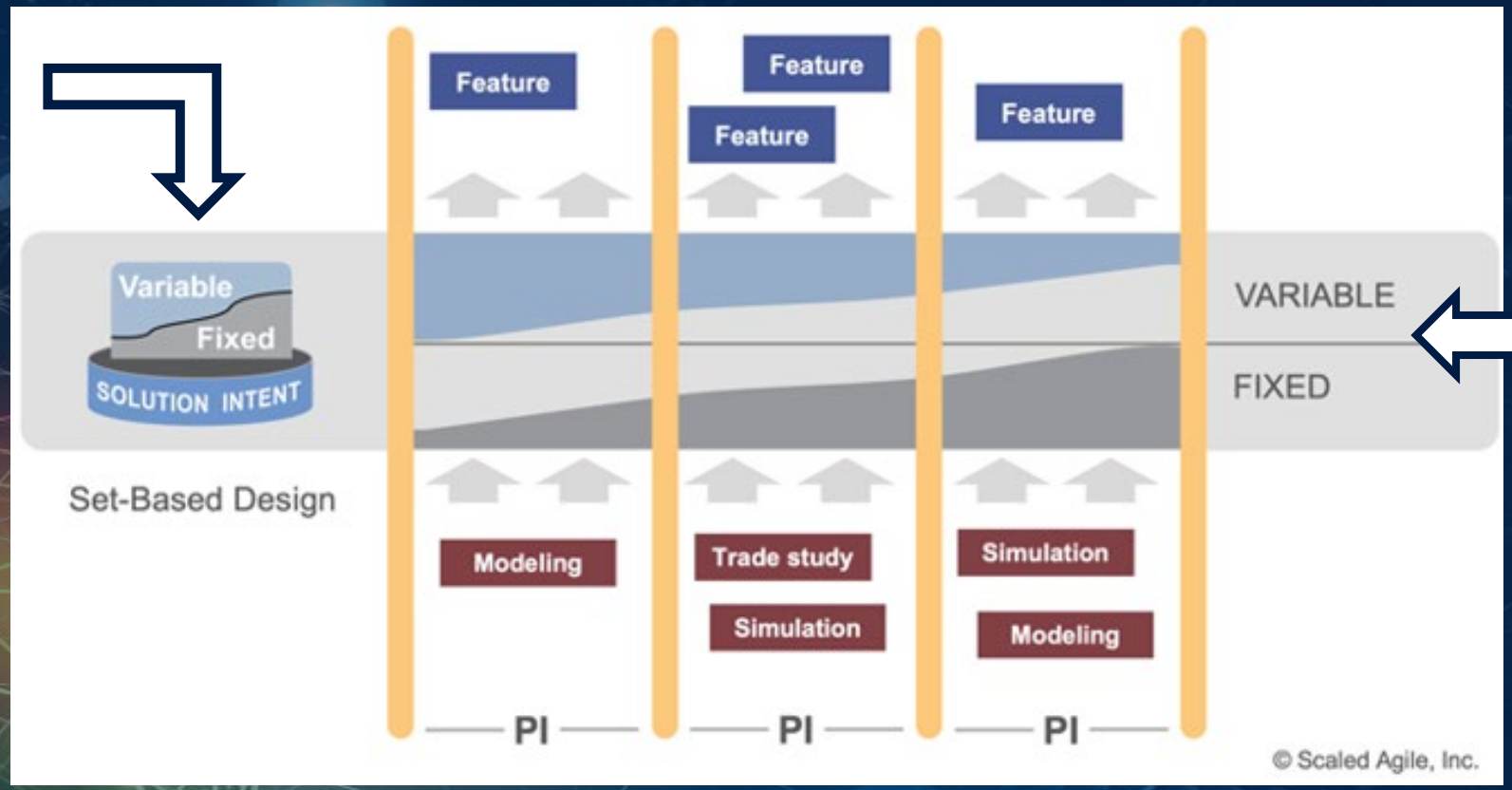
This is SOLUTION DEBT.

Solution debt is the extra effort (interest payments) required to overcome undesirable complexities in a solution and/or its supporting ecosystem - encountered during its analysis, design, development, or deployment.

Solution Debt Correlation with Solution Intent [28]

What is being built

How will it be built



Plateau of acceptable debt risk

Technical Discoveries

Solution Debt Decomposition & Common Symptoms [16]

Solution Debt Decomposition			Symptoms (non-exhaustive list)
Solution Debt	Business Debt	Management Debt	Inconistent views of roles/responsibilities, Unchecked resistance to change
		Supply Chain Debt	Reliance on single sources, Lack of visibility into supply sub-tiers
		Innovation Debt	Lack of differentiating capability, Unbalanced R&D portfolio
		Infrastructure Debt	Insufficient facilities or equipment, Lack of internal/external collaboration capability
		People Debt	Poor workforce morale, Missing skills or experience
		Process Debt	Ineffecient process, Undocumented process
		Service Debt	Ill-defined service level agreement, high service switching cost
		Social Debt	Strained interactions with stakeholders, Soiled reputation internally or externally
	Lifecycle Debt	Production Operation	Uncharacterized value streams, Inability to meet contracted volumes
		Architecture Debt	Unintended system behaviors, Lack of variation management
		Technical Data Package Debt	Incomplete specification, Inconsistent model methods and style
		Security Debt	Limited cyber recovery planning, Conflicting classification guides
		Obsolescence Debt	Surprise lifetime buys, Obsolete inventory
		Design Debt	Design for export not considered, Lack of design guidance
		User Interface Debt	Unnecessary cognitive load on user, Lack of realiable interpretation
		Requirement Debt	Unverifiable requirements, Incomplete traceability
		Code Debt	Duplicated code, Poor code metrics
		Defect Debt	Unmanaged backlog, Uncorrected major defects
		Test Debt	Lack of control and observability points, Unrepeatable test results
		Technology Debt	Ad-hoc technology adoption, Solution no longer effective in intended environment
		Build & Validation Debt	Lack of automated validation suite, Ill-defined dependencies
		Data Debt	Ad-hoc data controls, No data disposal plan

Digital Engineering Aspirations

- Facilitate better solutions, faster
- Navigate change
- Reduce risk
- Achieve greater mission assurance

Background Image by kjpargeter on Freepik

DE Capabilities Solve Problems, but Understand the Debt Potential (partial set of examples)

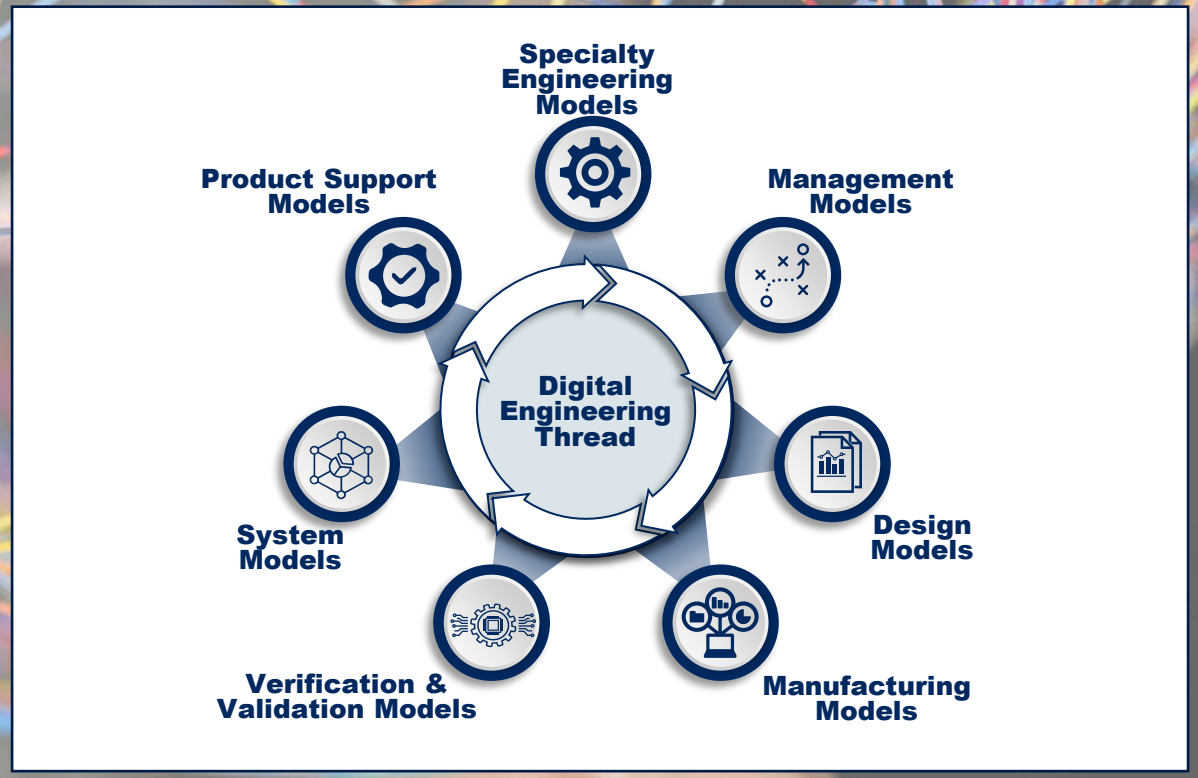
Solution Debt	Business Debt	Management Debt
		Supply Chain Debt
		Innovation Debt
		Infrastructure Debt
		People Debt
		Process Debt
		Service Debt
		Social Debt
	Lifecycle Debt	Production Operation
		Architecture Debt
		Technical Data Package Debt
		Security Debt
		Obsolescence Debt
		Design Debt
		User Interface Debt
		Requirement Debt
		Code Debt
		Defect Debt
		Test Debt
		Technology Debt
		Build & Validation Debt
		Data Debt

- Bespoke tool and repository integrations
- Lack of DE skills in the workforce
- Immature DE processes and methods
- Architecture connected throughout lifecycle
- More complete specification
- Robust design analysis spanning digital thread
- Early virtual integration
- Increased automation
- Connected, Authoritative Sources
- Bad Data Proliferation

DE capability:
- Mitigates debt
- Incurs debt

Example: Digital Engineering and Data Debt [12,13,14,22,23]

- Experian: The cost of bad data is 15% to 25% of revenue for most companies
- IBM: Businesses lost $3 trillion dollars per year due to bad data
- Gartner: Every year, poor data quality costs organizations an average $12.9 million



Digital threads connect data in a useful way, but also potentially 'super spread' bad data.

How does one evaluate debt? [15,21]

Impact

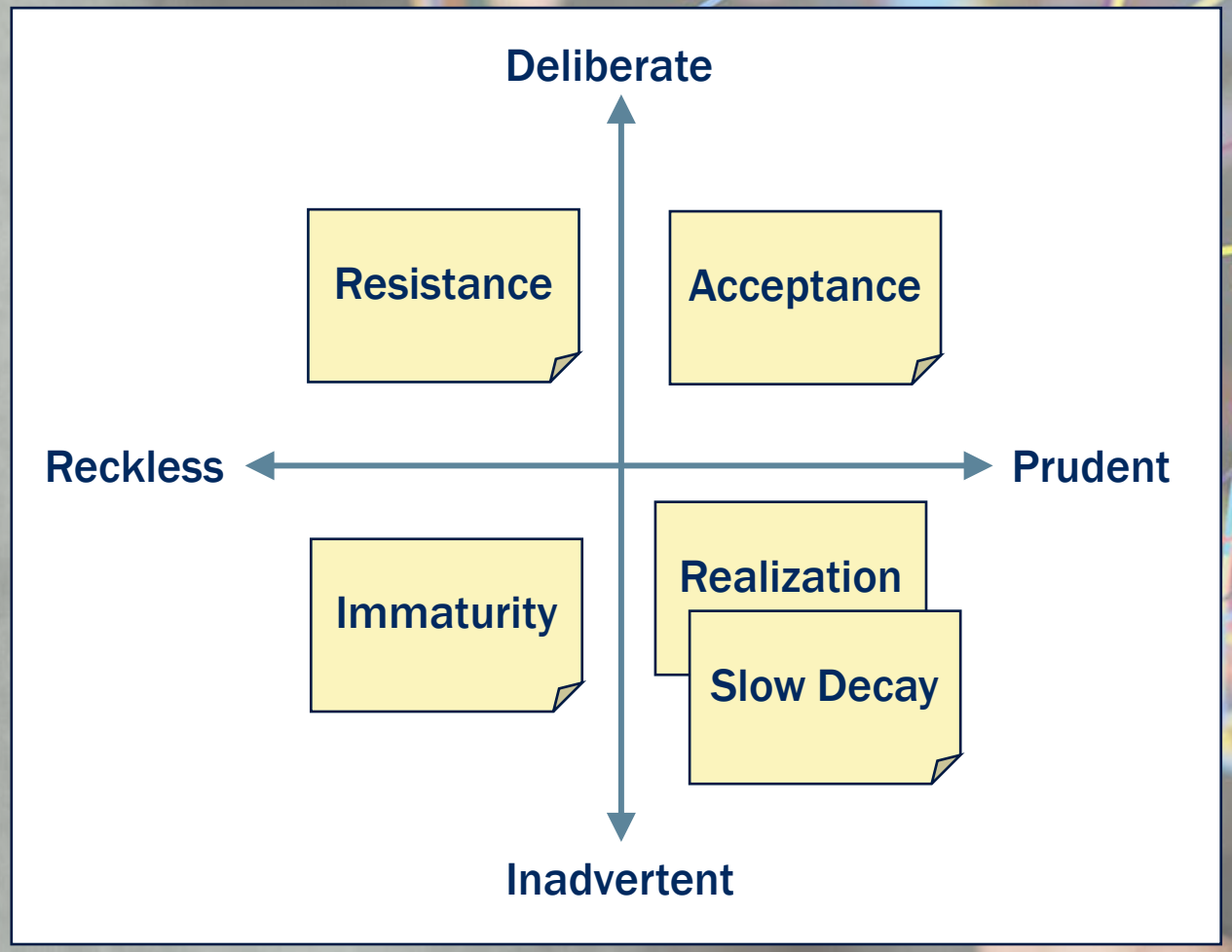
Fix Cost

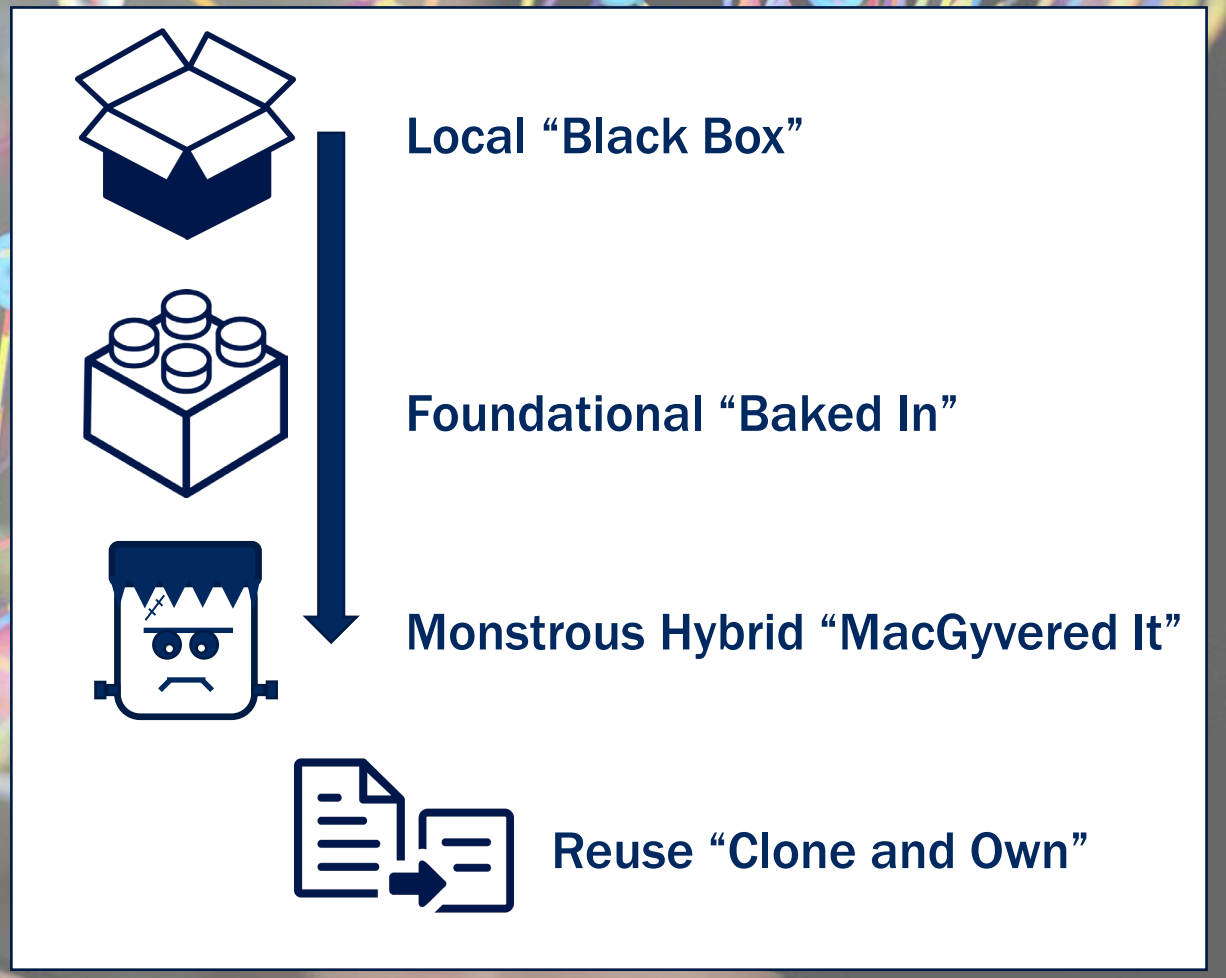
Contagion

Who pays?
(Directly?, Indirectly?)

Access Denied?
(No ability to impact)

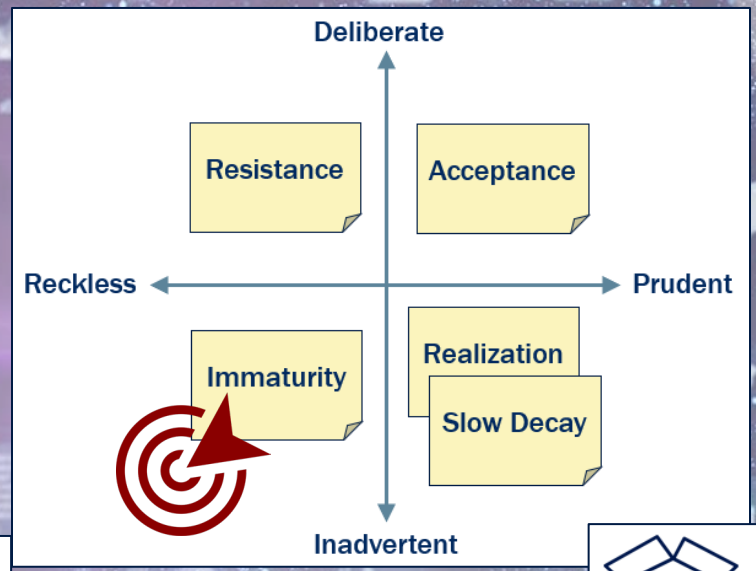
Are there common types of debt? [11,15]





Evaluate to Inform a Debt Reduction Plan

Impact	2/5
Fix Cost	3/5
Contagion	4/5



Who pays?
(Directly?, Indirectly?)

Oh no, my customer!

Access Denied?
(No ability to impact)

Yes, I need...

- Local "Black Box"
- Foundational "Baked In"
- Monstrous Hybrid "MacGyvered It"
- Reuse "Clone and Own"

Future Work

Expand and formalize the framework for action:

1. Formalize a solution debt taxonomy with definitions
2. Evaluate current program debt, using the techniques presented
3. Link evaluated program debt to risks, and explore linkages to governance which may ‘guardrail’ certain kinds of debt
4. Determine a program debt reduction plan and timeline
5. Execute the debt reduction plan

References

1. Skill Soft Course - Scrum Master: Scrum Master Service to Scrum Team (Colin Calnan)
2. https://dod.defense.gov/Portals/1/Documents/pubs/Section-901-FY-2017-NDAA-Report.pdf
3. https://discover.dtic.mil/wp-content/uploads/809-Panel-2019/Volume3/Sec809Panel_Vol3-Report_Jan2019_part-1_0509.pdf
4. https://aaf.dau.edu/
5. https://www.diu.mil/
6. https://www.gartner.com/smarterwithgartner/lack-of-skills-threatens-digital-transformation
7. https://hbr.org/2020/05/digital-transformation-is-about-talent-not-technology
8. https://media.defense.gov/2022/Mar/17/2002958406/-1/-1/1/SUMMARY-OF-THE-JOINT-ALL-DOMAIN-COMMAND-AND-CONTROL-STRATEGY.PDF
9. https://ac.cto.mil/digital_engineering/
10. https://www.incose.org/docs/default-source/se-vision/incose-se-vision-2035.pdf?sfvrsn=e32063c7_10
11. https://johnladley.com/a-bit-more-on-data-debt/
12. https://sloanreview.mit.edu/article/seizing-opportunity-in-data-quality/
13. https://venturebeat.com/datadecisionmakers/bad-data-a-3t-per-year-problem-with-a-solution
14. https://www.gartner.com/smarterwithgartner/how-to-improve-your-data-quality
15. https://technology.riotgames.com/news/taxonomy-tech-debt
16. https://ieeexplore.ieee.org/document/6974882
17. https://www.researchgate.net/figure/Technical-debt-taxonomy_fig1_319294348

References

18. https://www.techopedia.com/definition/27913/technical-debt
19. https://www.splunk.com/en_us/data-insider/what-is-tech-debt.html
20. https://www.gartner.com/en/documents/3989188#:~:text=Technology%20debt%20is%20the%20outstanding,design%20and%20development%20(see%20).
21. https://cognopia.com/are-your-customers-paying-your-data-debt/
22. https://clarkstonconsulting.com/insights/data-debt-impacts/
23. https://www.dataversity.net/data-debt-one-way-of-impacting-at-the-data-portfolio/
24. https://tedium.co/2016/07/19/y2k-hoarding-survivalists-freakout/
25. https://www.washingtonpost.com/outlook/2019/12/30/lessons-yk-years-later/
26. https://www.newsweek.com/why-do-we-buy-myth-y2k-166590
27. https://www.congress.gov/congressional-report/106th-congress/senate-report/10
28. https://www.scaledagileframework.com/solution-intent/

Thank you

For more information contact:

Dr. Douglas Orellana, Douglas.Orellana@ManTech.com

Dr. Heidi Davidz, Heidi.Davidz@ManTech.com