RT 193: Framework for Analyzing Versioning and Technical Debt

Sponsor: RDECOM CERDEC

By
Dr. Ye Yang, Dr. Jon Wade, Turki Aleyani, Patrick Stanton
Stevens Institute of Technology
10th Annual SERC Sponsor Research Review
November 8, 2018
FHI 360 CONFERENCE CENTER
1825 Connecticut Avenue NW, 8th Floor
Washington, DC 20009

www.sercuarc.org

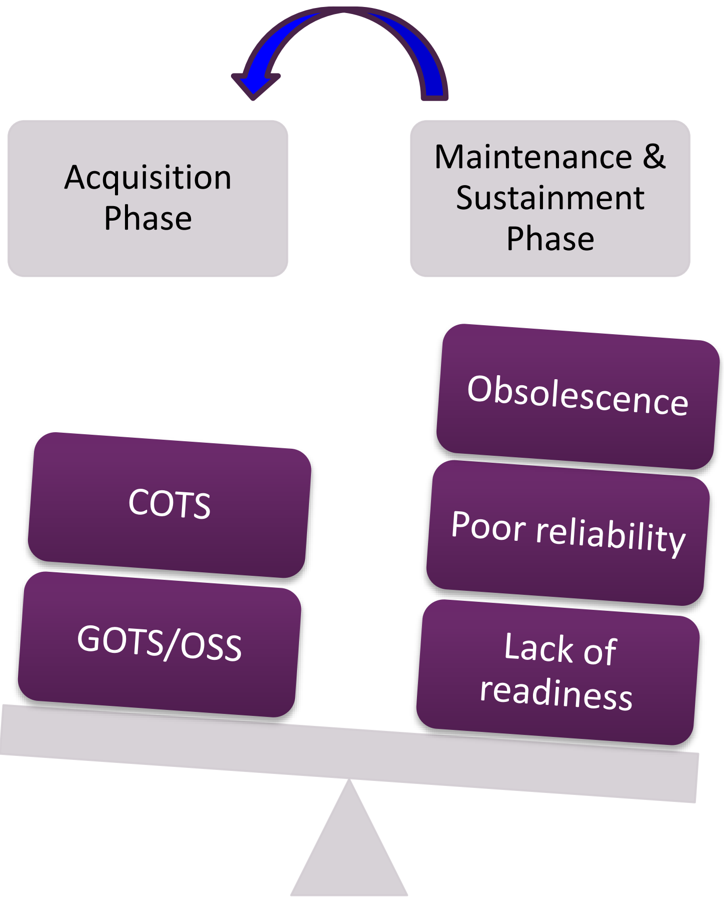
Background

- Obsolescence is a complex mix of engineering, economic, and business issues with many associated uncertainties.
- Obsolescence is the inevitable consequences of dependence on **COTS** components in many **Cyber-Physical-Systems (CPS)**
  - Long lead time of CPS, tightly-coupled components, shorter upgrade cycle of COTS, no control over COTS evolution, etc.

- *"Future Combat System had 153 relevant systems to deal with. If every one updated once a year, that would be a change every other day!"*
  ---- Barry Boehm, USC
- *"70 percent of electronics are obsolete prior to system fielding, and one component may become obsolete five to ten times during the weapon systems life cycle."* ---- Anthony Haynes, AMRDEC

Motivations

- **Problem Statement:**
  - COTS components in are increasingly imposing long-term management issues of many CPS systems
    - such as obsolescence, poor reliability, lack of readiness, and inability to be readily maintaining systems in an efficient and effective manner.
- **Motivations:**
  - Obsolescence is the consequence of COTS technical debt that can be possibly captured and managed in early CPS life cycle activities, i.e. COTS acquisition.
  - Increase awareness of COTS technical debt
  - Support early identification, assessment, and management of COTS technical debt



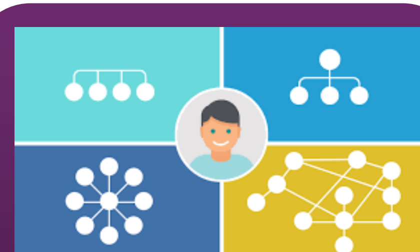
Research Framework

Literature Review

- Understanding trend in COTS related CPS Obsolescence studies

Mapping Framework

- Align existing MPTs
- Identify gap



COTS Technical Debt

- Taxonomy
- Meta attributes
- Simple Model

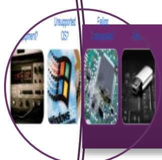
Literature Review

- Follow Kitchenham’s systematic literature review methodology
- Search Protocol
  - Keywords:
    - (“Technical debt” OR “Obsolescence”) AND
      (“COTS” OR “NDI” OR “GOTS” OR “Component*”) AND
      (“cyber physical system” OR “military systems” OR (“embedded systems”)
  - Databases
    - DMSMS; ACM Digital Library, IEEExplore, ScienceDirect, SpringerLink, Scopus, and Web of Science
- Search process
  - Three-round
  - Snowballing
- Results: a collection of 57 literatures included for further analysis

Literature Review Questions

RQ1: Trend in existing MPTs for COTS obsolescence?

RQ2: Types of data used?



RQ3: Sources of COTS obsolescence?

RQ4: Metrics for analyzing COTS obsolescence cost/risk?

RQ5: COTS obsolescence management approaches?

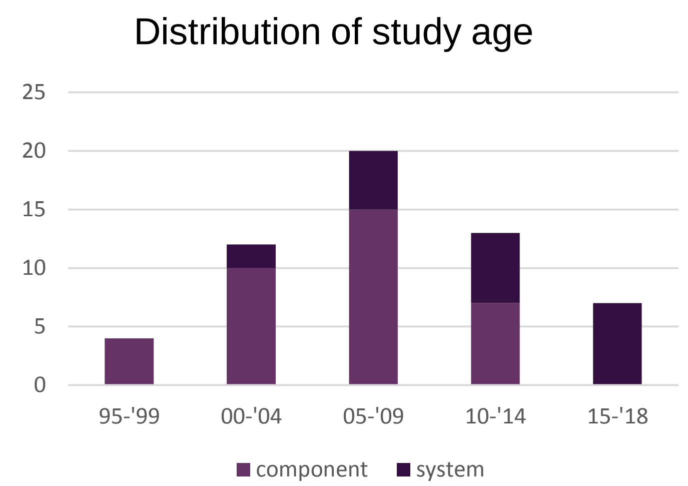
The review process focuses on extracting key information from individual study with regarding to the above review questions.

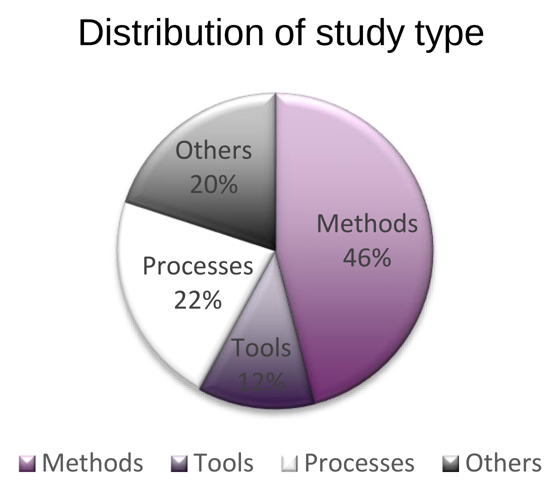
RQ1: What are the trend in existing MPTs for COTS obsolescence?

- Four categories to characterize current MPTs:
  - Type:
    - Methods
    - Processes
    - Tools
    - Others
  - Sector:
    - Academia
    - Industry
    - Government
    - Others
  - Targeted DoD Phases:
    - Materiel solution analysis
    - Technology maturation and risk reduction
    - Engineering and manufacturing development
    - Production and deployment
    - Operations and support
  - Granularity of obsolescence issue:
    - Component level
    - System level

Trends in Existing MPTs

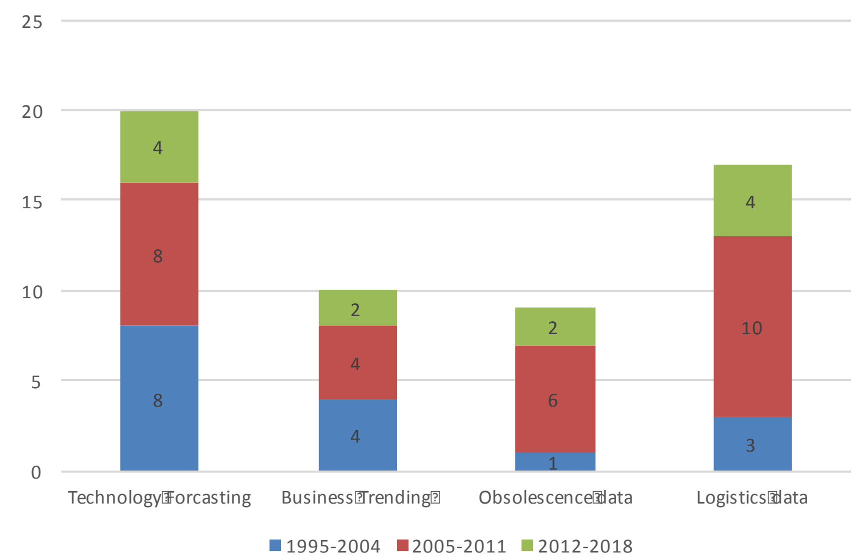
- Methods
  — Design Refresh; Life Time Buy; Last Time Buy; Substitution; Forecasting Model; VHDl-Based Model; Design Longevity Agreements, etc.
- Processes
  — Open source software products; Software Application programming Interfaces (API) and wrappers; After-market Supplier; Emulation/Cloning; Software Obsolescence Trigger Map
- Tools
  — COCOTS tool for estimating cost associated with COTS evaluation, tailoring, and integration; MOCA (mitigation of obsolescence cost analysis) tool; Total Obsolescence Management Capability Assessment Tool (TOMCAT); Component Information Management System





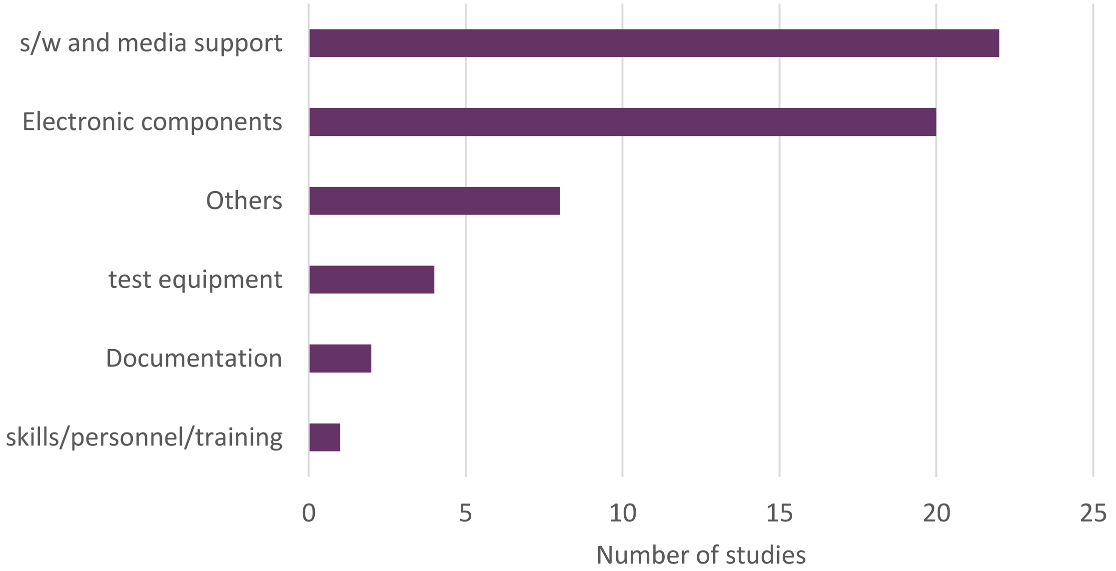
RQ2: Types of data used?

- Five categories of data
  - Technology forecasting: 20
    - E.g. High risk COTS/CCA (Circuit Card Assembly), OEM, BOM, contract incentives
  - Business Trending (Demand forecasting): 10
    - E.g. regression modelling to forecast business trend based on the obsolescence data and increased functionality of integrated circuits
  - Obsolescence data: 9
    - E.g. electronic/sw/media components
  - Logistics data: 17
    - E.g. DMSMS
  - Others: 19



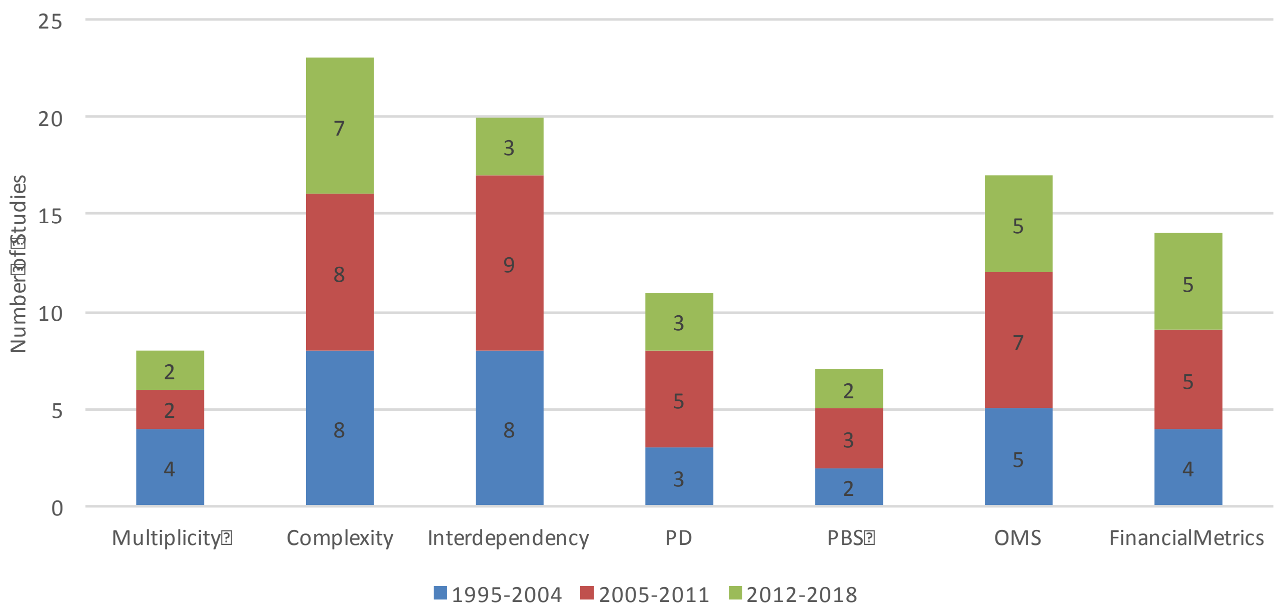
RQ3: Sources of COTS obsolescence?

- Six categories of COTS sources:
  - S/w and media support tooling: 22
    - E.g. operating system, ERP, database, etc.
  - Electronic components/Mechanical components: 20
    - E.g. EEE (electrical, electronic, mechanical) components, etc.
  - Test equipment: 4
  - Documentation: 2
  - Skills/personnel/training: 1
  - Others: 8



RQ4: Metrics for analyzing COTS obsolescence cost/risk?

- Seven categories of COTS metrics used in existing studies:
  - — Multiplicity (e.g. #of COTSs, #of components, etc.): 8 studies
  - — Complexity (e.g. system complexity, application complexity, Requalification complexity, etc.): 23
  - — Interdependency (e.g. Coupling level and package density, etc.): 20
  - — Platform diversity: 11
  - — PBS (product breakdown structure): 7
  - — OM strategy: 17
  - — Financial Metrics (e.g. RO, NPV, etc.): 14

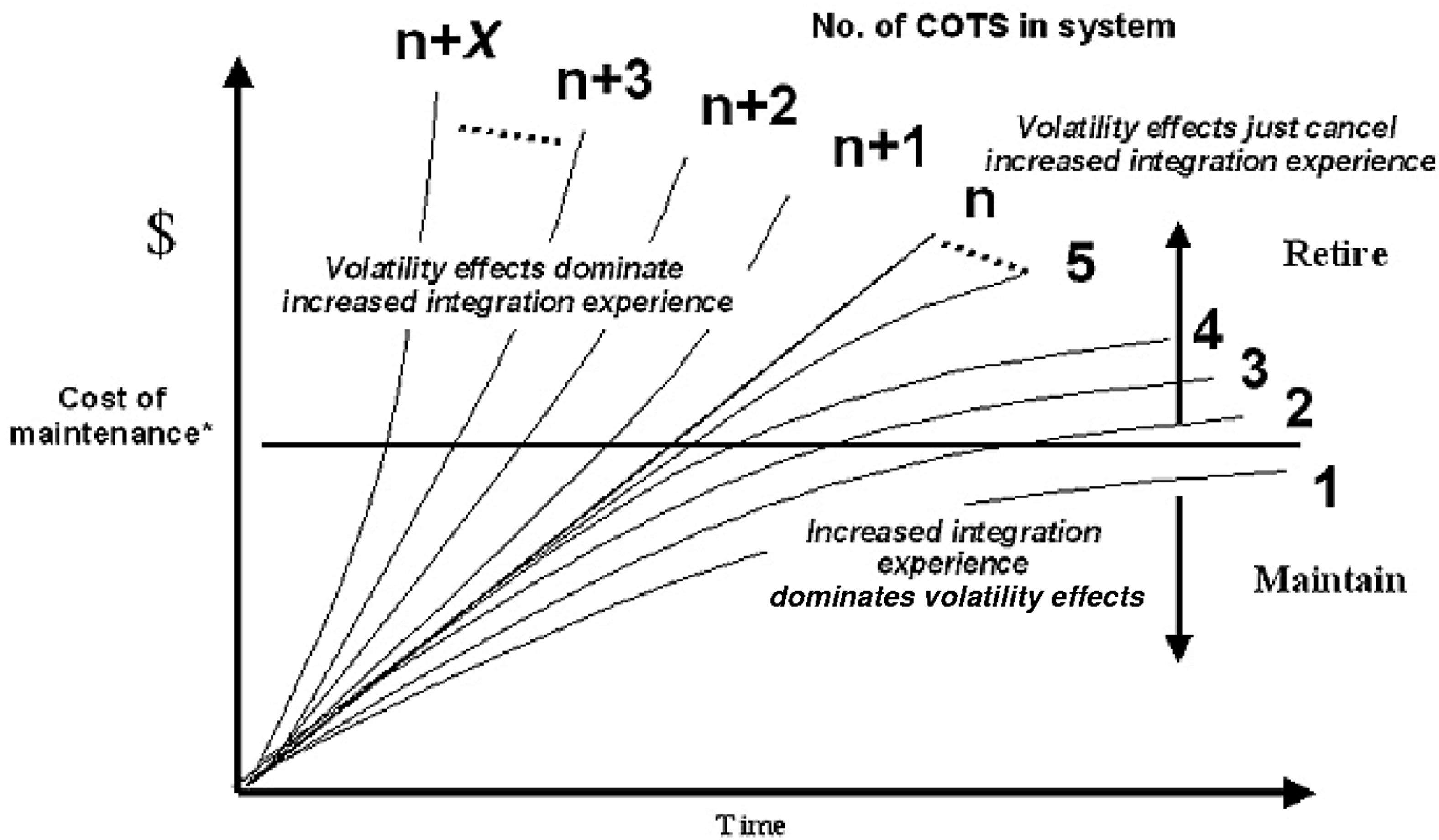


Example 1 - Cost metrics for requalification of air/safety critical components [Romero Rojo et al. 2012]

- **The cost metrics represent the non-recurring costs of resolving an obsolescence issue using each of the resolution approaches.**
  - during the contracted period within the in-service phase.

Obsolescence management approach	Integration level			
	Small	Medium	Large	Very large
Existing stock	£300	£300	£300	£300
Life time buy	£2,000	£2,000	£2,000	£2,000
Cannibalisation	£1,700	£2,500	£3,400	£4,500
Equivalent	£3,500	£3,500	£3,500	£3,500
Alternative	£10,100	£10,100	£15,200	£21,500
Authorised aftermarket	£13,000	£13,00	£19,800	£25,800
Emulation	£52,100	£193,000	£489,000	£2,690,000
Minor redesign	£50,100	£167,000	£244,000	£549,000
Major redesign	£250,000	£2,000,000	£3,400,000	£13,700,000

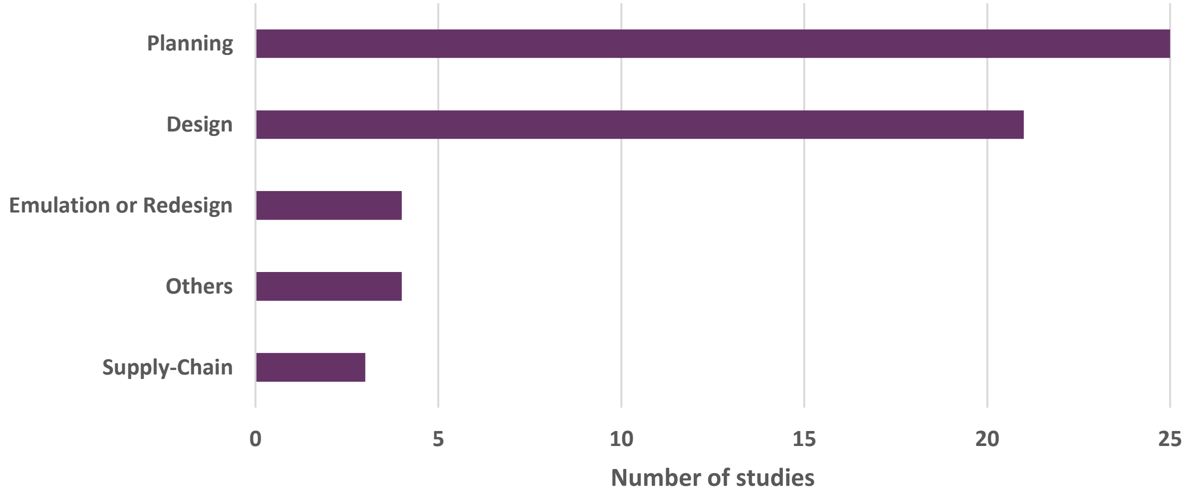
Example 2 - Economic Life Span of COTS-based Software Systems: the COTS-LIMO Model [Abt. Et al. 2000]



*Fn (synchronization, complexity of system, no. planned upgrades, etc.)

RQ5: COTS obsolescence management approaches?

- Three categories:
  - Strategic
    - Supply-chain: life-time buy and partnering agreement
  - Proactive
    - Design: open system architecture, modularity, use of multi-sourced components
    - Planning: obsolescence mgmt. plan, technology roadmap, monitoring tools
  - Reactive
    - Some components: last-time buy, cannibalization?
    - Form, fit & function(FFF) replacement (e.g. equivalent-component)
    - Emulation or redesign (e.g. use of state-of-art technology to replicate or redesign the component)



Discussion

- Proactive planning at system level is a largely overlooked topic and there is lack of study
- Opportunity: COTS Technical Debt Identification
  - Utilization existing OM MPTs must be strategically coupled and/or replaced with capabilities in the acquisition time, e.g.:
    - Capture interdependencies of COTS components in CPS systems;
    - Identify "technical debt" items associated with COTS decisions;
    - Predict the effects of COTS technical debt items on the system across its system life cycle;
    - Make informed technical decisions associated with COTS usage.

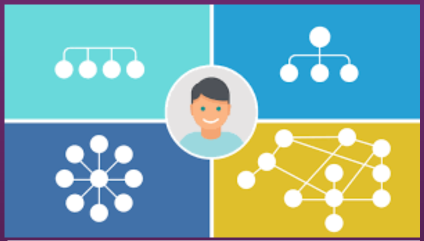
Research Framework

Literature Review

- Understanding trend in COTS related CPS Obsolescence studies

Mapping Framework

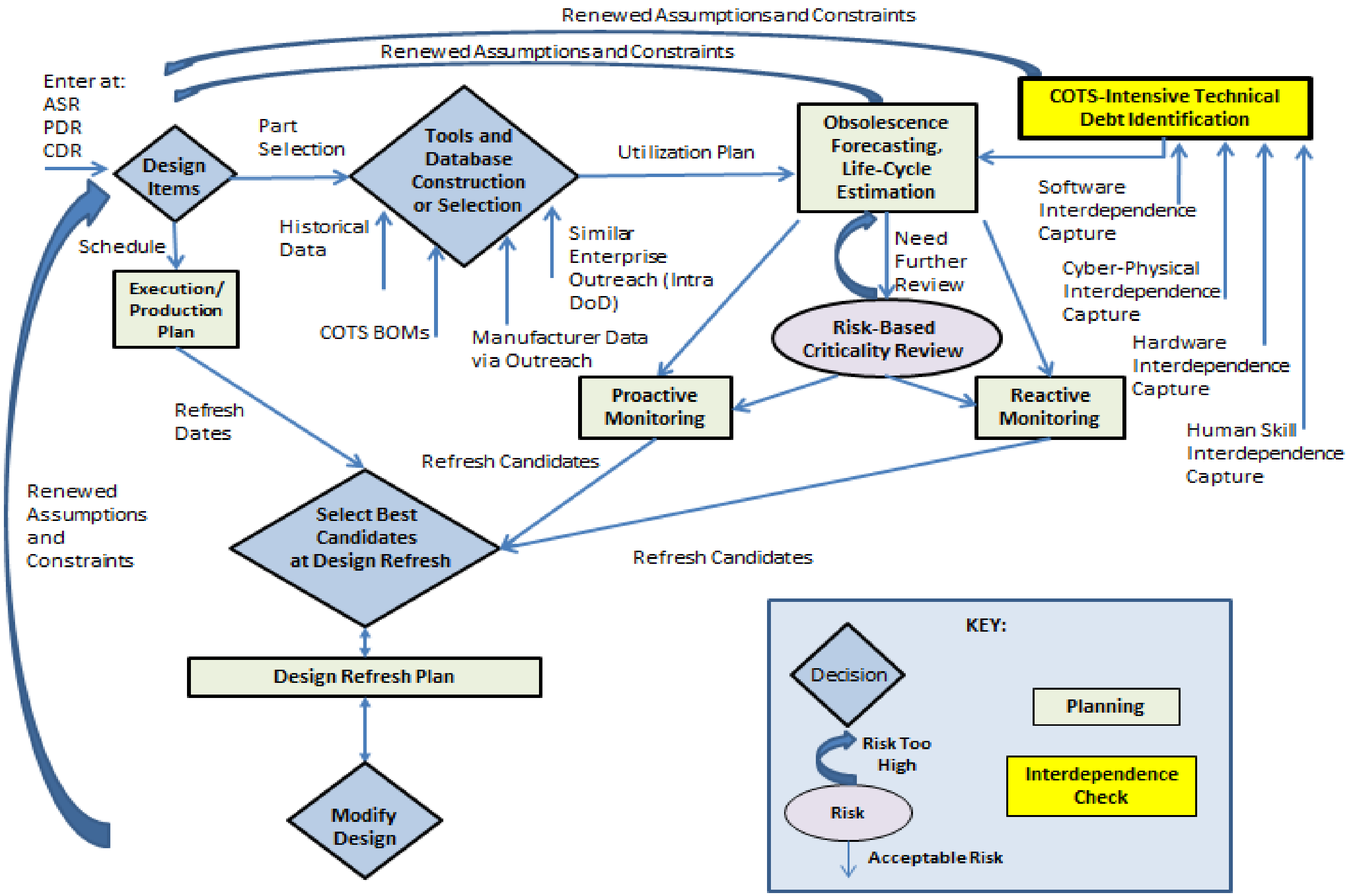
- Align existing MPTs
- Identify gap



COTS Technical Debt

- Taxonomy
- Meta attributes
- Simple Model

Mapping Framework



Hybrid Flow of Obsolescence Risk and COTS Technical Debt Management

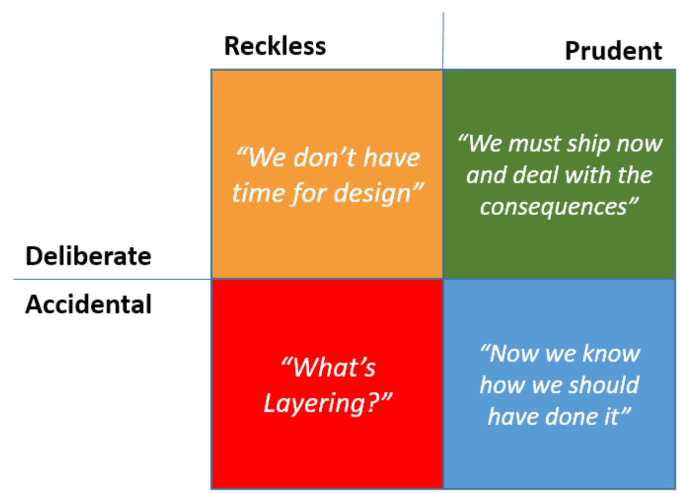
The Notion of Technical Debt

- Originated in software engineering field, coined by Ward Cunningham in 1992
  - Immature work, compromising in one dimension in order to get benefits in other dimensions
  - Initially concerning ”refactoring” at code level (i.e. implementation) in agile software development

- Evolved to span across all life cycle phases
  - a metaphor reflecting technical compromises that can yield short-term benefit but may hurt the long-term health of a software system

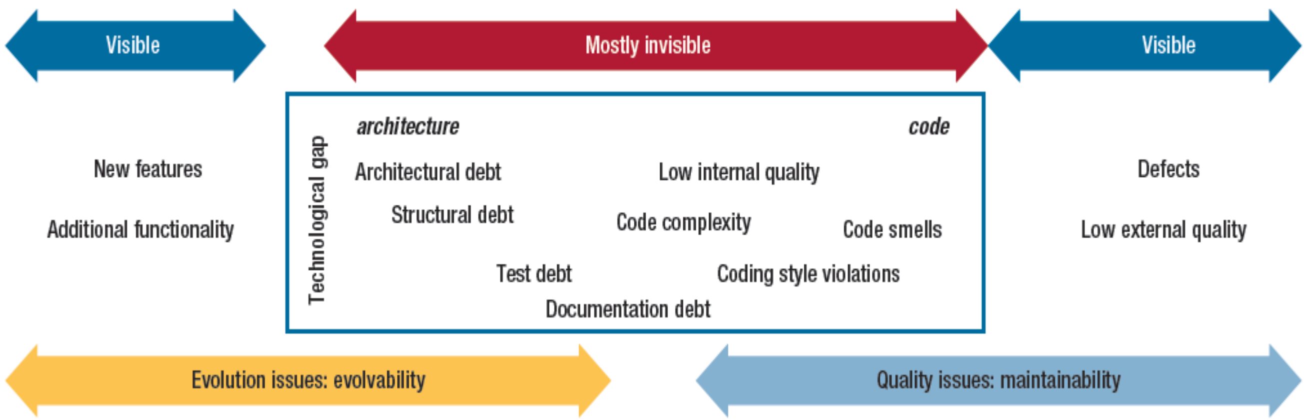
- Technical Debt Quadrants [Martin Fowler, 2009]

Technical Debt Quadrants



What Constitutes Technical Debt?

- Technical Debt Landscape (Ozkaya, Nord, Kruchten, 2012)
  - Differentiate visible elements from invisible elements

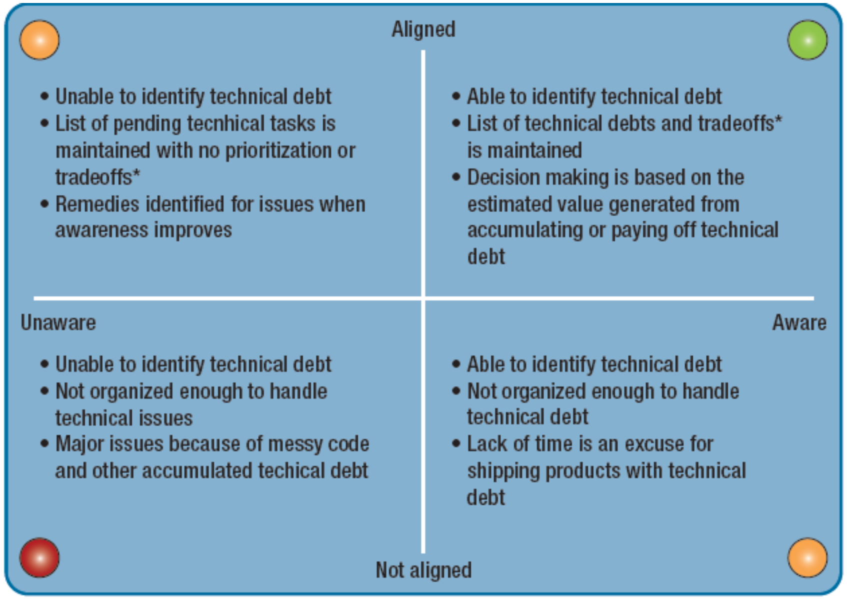


  - Propose to limit debt to the invisible elements
    - Four colors in a backlog

	Visible	Invisible
Positive value	New features Added functionality	Architectural, structural features
Negative value	Defects	Technical debt

Some Existing Taxonomies on Technical Debt

- Rubin's Taxonomy
  - Context: within Agile team
    - Naïve technical debt
    - Unavoidable technical debt
    - Strategic technical debt
- Clark's Taxonomy
  - Context: Riot Games (*League of Legends)*
    - Local debt
    - MacGyver debt
    - Foundational debt
    - Data debt
- Bavani's Taxonomy
  - Context: distributed teams & agile testing
    - Degree of awareness of technical debt across distributed teams
    - Degree of alignment in managing technical debt across distributed teams



*Tradeoff can be an alternate approach to lessen the impact of technical issues or debt

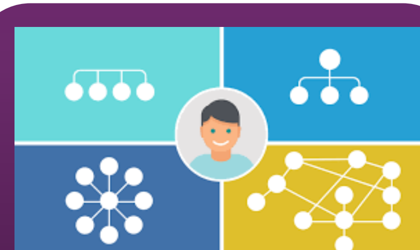
Research Framework

Literature Review

- Understanding trend in COTS related CPS Obsolescence studies

Mapping Framework

- Align existing MPTs
- Identify gap



COTS Technical Debt

- Taxonomy
- Meta attributes
- Simple Model

SYSTEMS ENGINEERING RESEARCH CENTER

Technical Debt as A Metaphor for Predicting COTS Obsolescence

COTS Benefits	COTS Implications	COTS "Technical Debt"
Available now, earlier payback	Licensing and procurement delays	N/A
Avoids expensive development & maintenance	Up front license fees	N/A
Predictable license costs & performance	Recurring maintenance fees	Yes. Incurred COTS upgrading cost and system re-evaluation/re-testing cost
Rich in functionality	Reliability often unknown/ inadequate; Unnecessary features compromise usability, security, performance	Yes. Incurred cost to take care of functional/non-functional requirement mismatch and additional verification & validation
Broadly used, mature technology	Functionality, efficiency constraints	Yes. Incurred cost to tailor to specific CPS context; increased limitation over system evolution
Frequent upgrades often anticipate organization's needs	No control over upgrades/maintenance	Yes. Increased obsolescence risk due to life cycle mismatch between CPS system and COTS components
Dedicated support organization	Dependency on vendor	Yes. Increased obsolescence risk due to documentation and support dependency
Hardware/software independence	Integration not always trivial; incompatibilities among different COTS	Yes. Incurred cost to evaluate and enhance COTS interoperability in COTS-intensive CPS.
Tracks technology trends	Synchronizing multiple-vendor upgrades	Yes. Increased obsolescence risk due to life cycle mismatch between CPS system and COTS components

COTS TD Taxonomy in CPS Context

COTS TD Category	Description	Analogy to existing work
Function	The degree of functionality mismatch between COTS capabilities and system needs.	Local TD; Data TD
Performance	The degree of mismatches between COTS capabilities and system needs, w.r.t. quality/extra-functional properties such as: (1) Reliability – mainly of hardware; (2) Safety assurance – of software and hardware; (3) Performance in terms of e.g. bandwidth, processing capability, memory etc.	MacGyver TD; Data TD
Interoperability	The degree of interface/ assumption mismatches among various interdependent COTS components, as well as among COTS and system custom components.	MacGyver TD; Data TD
Configuration Version	CPS configuration version planning needs to address solution availability plan. Greater tendency of COTS version upgrade/refresh may lead to more obsolete COTS.	Unavoidable TD; Local TD; MacGyver TD; Foundational TD; Data TD
Documentation & Support	Lack of documentation and vendor support will seriously impact on issue resolution related to obsolete COTS.	Unavoidable; Data TD
System Evolution Limitations	Requirements imposed by COTS may place great limitation on system evolution.	Unavoidable TD; Foundational TD; Data TD
Organic	People-centric perspective of TD focusing on organizational decision-making, behaviors, and practices associated with those personnel responsible for introductions of new technologies & systems and/or the sustainment of existing systems	Local TD; Naïve TD; Strategic TD

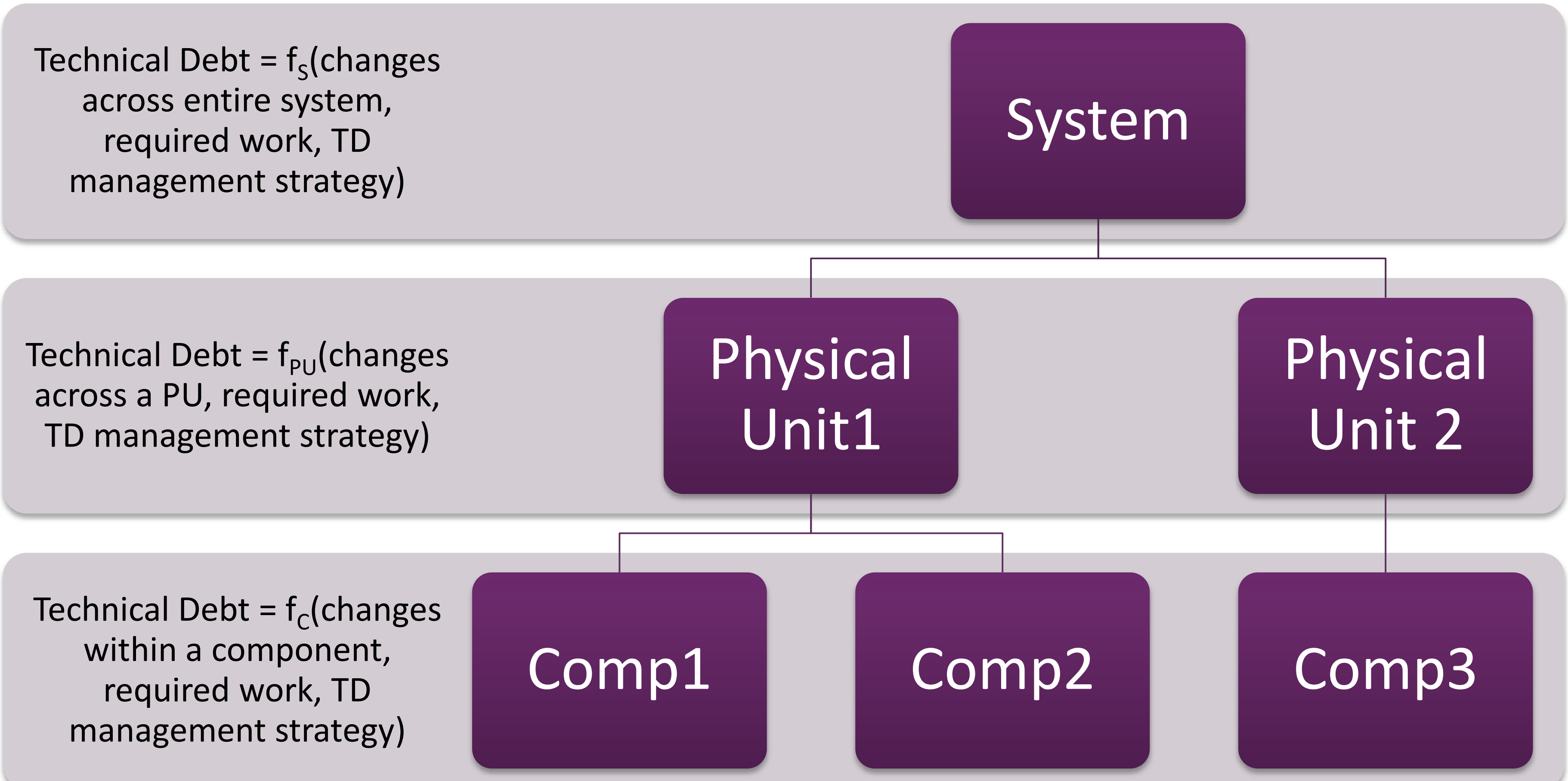
Attributes For Representing A COTS TD Item

Attribute	Description
ID	A unique identifier for the COTS TD item.
Name	The name of a specific COTS TD item
Location	The location of the identified COTS TD item, e.g. the name of the COTS(s) with which it is associated.
Accountable Party	The party responsible to repay the COTS TD item, e.g. COTS vendor, integration team, program office, specific organization. This identifies the "accountable" debt-holder for the liability. The Accountable Party is identified at the start of a new design/development/ modernization effort, and can assign TD "tracking" and "maintenance of TD visibility" within its span of authority/control.
Type	The COTS TD type that the COTS TD item is classified into.
Description	General information on the COTS TD item.
Open date/time	The specific date/time when the COTS TD is identified.
Principle	The estimated cost of repaying the COTS TD item.
Interest amount	The estimated extra cost of tolerating the COTS TD item.
Interest probability	The probability that the interest for the COTS TD item needs to be repaid.
Contagion	The degree of spreading of the COTS TD item through the interfaces with other system components, if this TD is allowed to continue to exist.
Context	A certain implementation context of a specific COTS TD item
Propagation rule	How the COTS TD item impacts the related parts of the CPS system
Intentionality	Is the COTS TD item Intentionally or unintentionally incurred?

COTS TD Management Activities

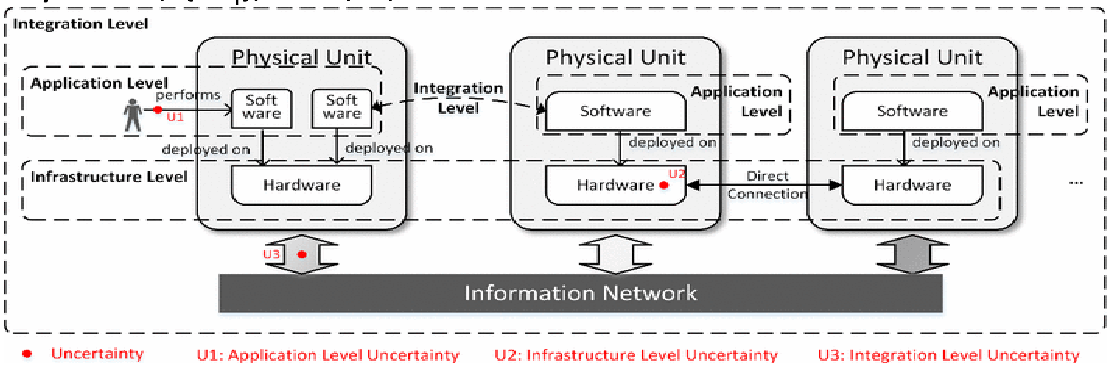
TDM Activity	Description/Example	Techniques	Example metrics
TD identification	Detects TD caused by intentional or unintentional technical decisions	Static code analysis; dependency analysis; checklist	Violations of coding rules, lack of tests; static code metrics,
TD measurement	Quantifies the benefit and cost of known TD in a system through estimation techniques	Expert Estimation; estimation models; cost categorization; solution comparison	code metrics; operational metrics; ROI; Cost-benefit ratio; Real options
TD prioritization	Ranks identified TD items according to predefined rules, which is to be repaid first, and which can be tolerated until later releases.	Cost benefit analysis; High remediation cost first; Portfolio approach; High interest first	Portfolio approach considering TD items along with other new functionalities and bugs as risk and investment opportunities.
TD prevention	Aims to prevent certain TD from being incurred.	Development process improvement; design decision support; lifecycle cost planning; human factor analysis	Improve process to prevent certain type of TD; evaluate and choose candidate solutions with less potential TD
TD monitoring	Watches the change of cost and benefit of unresolved TD over time	Threshold-based; Planned check; TD propagation tracking; TD plot; TD monitor with quality attribute focus	Define threshold for quality metrics, and issue warnings if threshold is not met.
TD repayment	Resolves or mitigates TD	Reengineering, rewriting; refactoring; bug fixing; fault tolerant; repackaging; automation	Make changes to the code, design, or architecture of the software system without altering external behavior, in order to improve internal quality.
TD representation/ documentation	Provides ways to represent and codify TD in a uniform manner to address concerns of particular stakeholder	Various format of representing TD items.	Example TD data fields: ID, Location, Responsible / author, Type, Description, date /Time, principle, interest amount, interest probability, relation to other TD, context, propagation rule, intentionality
TD communication	Makes identified TD visible to stakeholders so that it can be discussed and further managed.	TD dashboard; backlog; dependency visualization; code metric visualization; TD list; TD propagation visualization	Dashboard or other visualization tool displaying undesirable dependencies, e.g. overly complex dependencies between system components

Hierarchical View of a Simple Technical Debt Model for COTS-Intensive CPS



Modeling COTS-intensive CPS

- COTS-intensive CPS
  - — A set of physical units, i.e. subsystems, $\{SS_i\}$, i=1, 2, …. M
  - — Attributes:
    - ○ Budget, schedule
    - ○ %reqt's covered by COTS
    - ○ Planned upgrade cycle
    - ○ Acquisition cost
    - ○ COTS technical debt



- Dependency matrix
  - — Interface requirements among all components

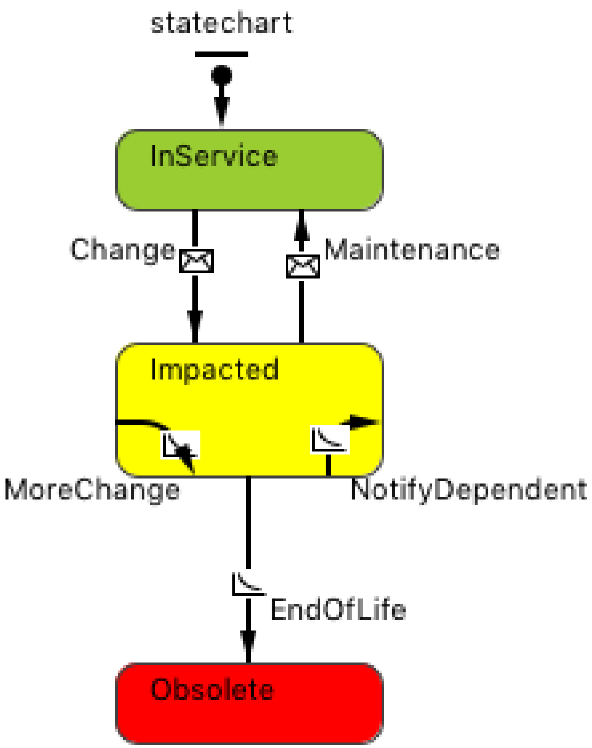
- Multi-Agent Models
  - — Each physical unit, SS_i
    - ○ A set of hardware and/or software components, $\{C_{ij}\}$, j=1, 2, …. n_i
    - ○ Type: Application, Infrastructure, Network, other
  - — Each component, C_{ij}
    - ○ Attributes: %reqt's gap; acquisition cost, upgrade cycle, upgrading cost
    - ○ Type: COTS h/w, COTS s/w, custom h/w, custom s/w, other

Modeling COTS Configuration Version Technical Debt

- Discrete Event Model
  - COTS change events
    - COTS change:
      - Upgrade cycle: Probabilistic distribution function: e.g. [6month, 12month]
      - Change ratio: random variable {0, 1), larger number indicating greater portion of COTS is changed
  - TD management actions
    - TD Principal Measurement
      - Component level: f_C(change ratio, required work, TD reduction strategy)
      - Physical Unit level: f_{PU}(changes across a PU, required work, TD reduction strategy)
      - System level: f_S(changes across entire system, required work, TD reduction strategy)
    - TD Reduction strategies
      - 0: no work
      - 1: upgrade every version
      - 2: upgrade every other version
      - 3: upgrade until end-of-life
    - TD Dynamic Forecasting
      - f(TD principal, probablity of TD interest, TD interest amount, t)

COTS Change Propagation and Change Impact Modeling

- COTS Change Impact Analysis
  - Dependency matrix
    - Coupling rate
  - State transition model
    - InService
    - Impacted
    - Obsolete



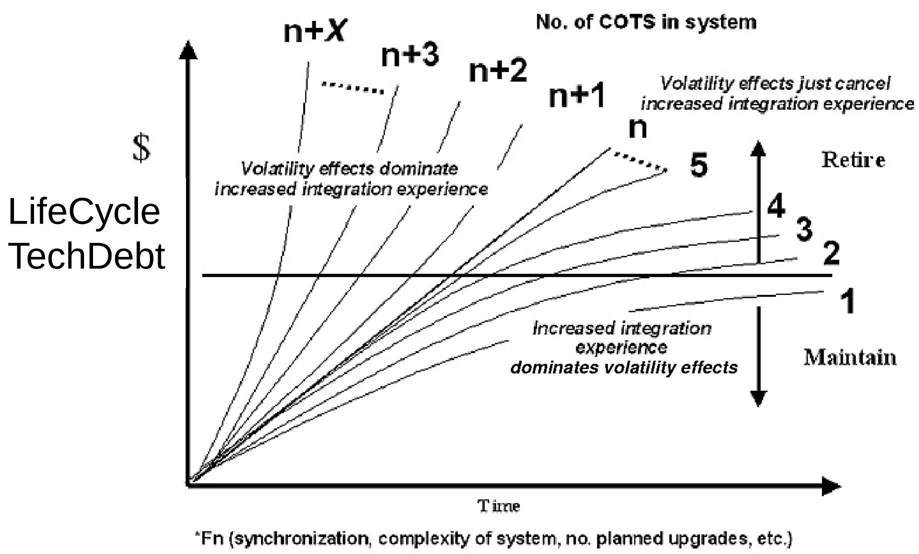
COTS TD Interest Probability based on COCOTS Model

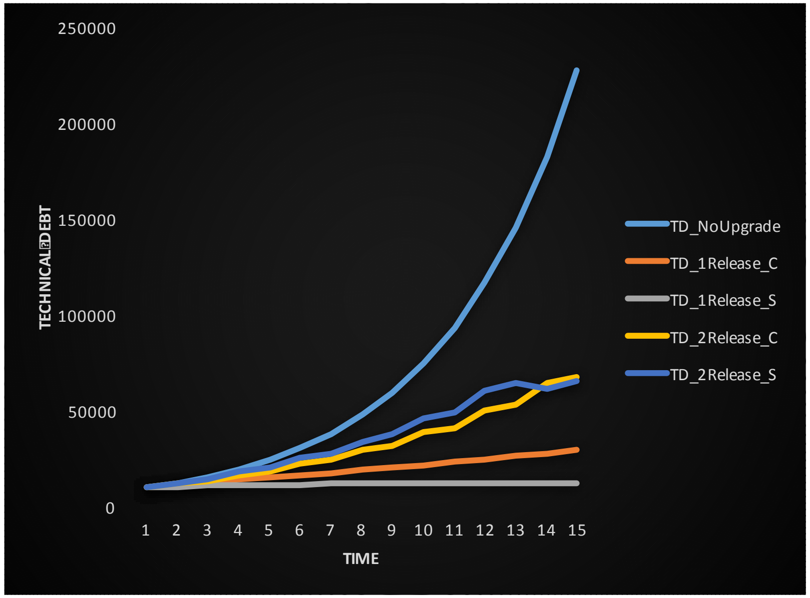
- COCOTS is an effort/cost estimation model for COTS integration, developed at USC
  - — 3 submodels: COTS assessment, tailoring, integration
  - — 15 cost drivers: COTS integrator, COTS vendor, system
- Extension with 3 additional security drivers
  - — Required system EAL level
  - — COTS certified EAL level
  - — Degree of unused COTS features
- COCOTS Risk Analyzer
  - — Identify COTS integration risk from cost driver inputs
    - ○ A pair of cost drivers with two opposite extreme rating levels, e.g.
      - — very high system complexity vs. very low COTS product maturity,
      - — very high system complexity vs. very low COTS integrator capability
  - — Knowledge base of 24 rules

	Size	AAREN	ACIEP	ACIPC	AXCIP	APCON	ACPMT	ACSEW	APCPX	ACPPS	ACPTD	ACREL	AACPX	ACPER	ASPRT	APEAL	ACEAL	ACPUF	Productivity Range
SIZE							1		1			1							
AAREN(Application Architectural Engineering)				1	1		1		1			1	1	1	1				2.09
ACIEP(COTS Integrator Experience with Product)									1										1.79
ACIPC(COTS Integrator Personnel Capability)					1	1			1		1	1	1						2.58
AXCIP(Integrator Experience with COTS Integration Processes)						1	1												1.42
APCON(Integrator Personnel Continuity)											1								2.51
ACPMT(COTS Product Maturity)										1		1							2.1
ACSEW(COTS Supplier Extension Willingness)																			1.22
APCPX(COTS Product Interface Complexity)													1						1.8
ACPPS(COTS Supplier Produce Support)																			1.48
ACPTD(COTS Supplier Provided Training and Documentation)																			1.43
ACREL(Constraints on Application System/Subsystem Reliability)																			1.48
AACPX(Application Interface Complexity)																			1.69
ACPER(Constraints on COTS Technical Performance)																			1.22
ASPRT(Application System Portability)																			1.14
APEAL (Application Evaluated Assurance Level)																	1		2
ACEAL (COTS Evaluated Assurance Level)																			2
ACPUF (percentage of COTS' unused features)																			2
TD Risk Probability:																			
>=50%																			
[40%, 50%)																			
[20%, 40%)																			

Examples of Decision Scenario Simulation

- Scenario 1: Selecting different COTS-based solutions
- Scenario 1: Dynamics of TD aggregation and reduction





Conclusions and Future Directions

- Conclusions
  - Compelling and critical need for a Systems Engineering technical debt metaphor grows
  - The notions of COTS technical debts will help to inform COTS decision making practices in the acquisition process to avoid unaffordable obsolescence issues particularly in the sustainment phase
  - Taxonomy of COTS-related technical debt can support early identification, communication, and assessment of obsolescence risks in CPS system engineering life cycles

- Future directions:
  - Map major obsolescence issues in existing case studies to the proposed COTS TD taxonomy
  - Modelling and Simulation of COTS changes and impact on technical debt aggregation within CPS
  - Align COTS TD management techniques and align with existing acquisition activities

Thank you!