Practical Software and Systems Measurement

Objective Information for Decision Makers

What Does Technical Debt Mean at a System Level

2 August 2012

Bob Epps, Lockheed Martin

What Does Technical Debt Mean at a System Level

- *Software Technical Debt has been an area of increasing focus and discussion, but we also believe it has application to Systems Engineering, as well.*
- *The intent of the workshop is to understand the scope of Technical Debt and what it means with respect to Systems Engineering.*

Objectives of the Workshop

- *Identification and Management of Technical Debt in System Development and Operation & Maintenance (O&M) phases of the System Lifecycle.*
- *Discuss how the Decisions made during the System Lifecycle influence the level of Technical Debt within the System.*
- *Evaluation of Technical Debt's impact to the Operation & Maintenance (O&M) phase.*
- *Discussion measurement/management of Technical Debt*

Workshop Format

- Agenda
  - Management of Technical Debt- Steve McConnell
  - Technical Debt Observations- Jim Highsmith
  - Types of Debt- Chris Sterling
  - Workshop Exercise # 1-Identifying Technical Debt
  - Break
  - Management of Architectural Debt- Ipek Ozkapa
  - Workshop Exercise # 2- Architecture Technical Debt
  - Workshop Exercise # 3- System Design Technical Debt
  - Break
  - Operations & Maintenance(O&M) Technical Debt
  - Measurement /Management of Technical Debt
  - Workshop Summary/Action Plan

Workshop Format

- *Techniques that will be used*
  - *We would look at the terms and concepts from Software Technical Debt and then compare and contrast as needed to ensure consistency, when adopted for Systems Engineering. The intent is to build from the existing knowledge rather than re-invent the wheel.*

Workshop Background

- *PSM history in this area*
  - *2011 Keynote Address: Cheryl McIntyre, Lockheed Martin*
  - *2011 Workshop: Bob Epps/Garry Roedler, Lockheed Martin*

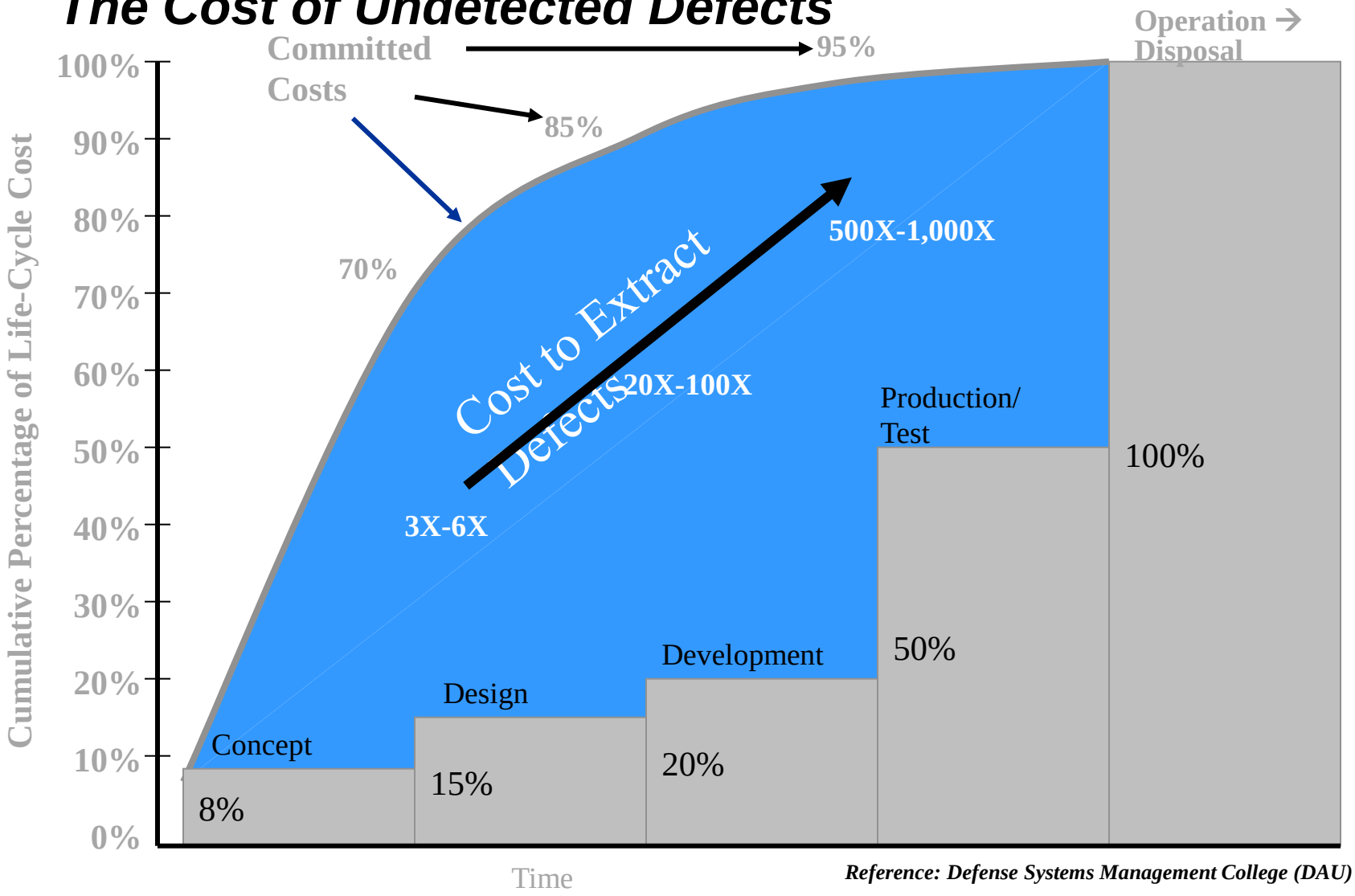
- *Where we're heading*
  - *Defining Technical Debt influence throughout the System Lifecycle*
  - *Measurement and Management of Technical Debt*

- *Issues, questions, and topics*
  - *Impact of Technical Debt on System Development phase, O&M phase & System Affordability*

Intended Output

- *Collaboration with other industry associations(NDIA, INCOSE, etc.) on a white paper on the topic of Systems Engineering Technical Debt.*

The Cost of Undetected Defects



Reference: Defense Systems Management College (DAU)

Technical Debt

"Management of Technical Debt", Steve McConnell

- " 'Technical Debt' refers to the delayed technical work that is incurred when technical short cuts are taken, usually in pursuit of calendar driven software schedules. Just like financial debt, some technical debt can serve valuable business purposes. Other technical debts are simply counter productive. The ability to take on debt safely, track their debt, manage their debt and pay down their debt varies among organizations. Explicit decision making before taking on debt and more explicit tracking of debt are advised"

Technical Debt

"Management of Technical Debt". Steve McConnell

- " 'Technical Debt' refers to the delayed technical work that is incurred when technical short cuts are taken, usually in pursuit of calendar driven software schedules. Just like financial debt, some technical debt can serve valuable business purposes. Other technical debts are simply counter productive. The ability to take on debt safely, track their debt, manage their debt and pay down their debt varies among organizations. Explicit decision making before taking on debt and more explicit tracking of debt are advised"

Technical Debt

"Management of Technical Debt" Steve McConnell

- " The term 'technical debt' was coined by Ward Cunningham to describe the obligation that a software organization incurs when it chooses a design or construction approach that's expedient in the short term but that increases complexity and is most costly in the long term"
- There are two kinds of technical debt
  - Type I, Debt incurred unintentionally
    - Inexperienced individuals produces error prone results
  - Type II, Debt incurred intentionally
    - Conscious decision to optimize for the "present" rather than the "future"

Technical Debt

"Management of Technical Debt", Steve McConnell

- Type II, Debt incurred intentionally

"Short-Term" Debt (Type II.A)

- A company takes on a short term debt when it has the money; it just does not have it now.
- Short term debt is expected to be paid off frequently
- Focused Short-Term Debt(Type II.A.1)
- Unfocused Short-Term Debt (Type II.A.2)
  - Should be avoided

"Long-Term" Debt(Type II.B)

- A company takes on strategically and proactively
- Primary rationale is that the development work "today" is seen as more expensive than the cost in the future.
- Example:
  - Responding to "Time to Market" pressures
  - Preservation of Startup capital
  - Delaying Development expense

Debt Service

- The "interest" charged for incurring the debt

Technical Debt

"Management of Technical Debt", Steve McConnell

Summary of Kinds of Debt

Non Debt

Features backlog, deferred features, cut features, and so on. Not all incomplete work is debt. These are not debt because they do not require interest payments

Debt

I. Unintentional Debt. Debt incurred unintentionally due to low quality

II. Intentional Debt. Debt incurred intentionally

II.A Short-Term Debt. Short Term Debt, usually incurred reactively, for tactical reasons

II.A.1 Focused Short Term Debt. Individually identifiable shortcuts(like a car loan)

II.A.2 Unfocused Short-Term Debt. Numerous tiny shortcuts(like a credit card)

II.B Long-Term Debt. Long-term debt, usually incurred proactively, for strategic reasons

Technical Debt

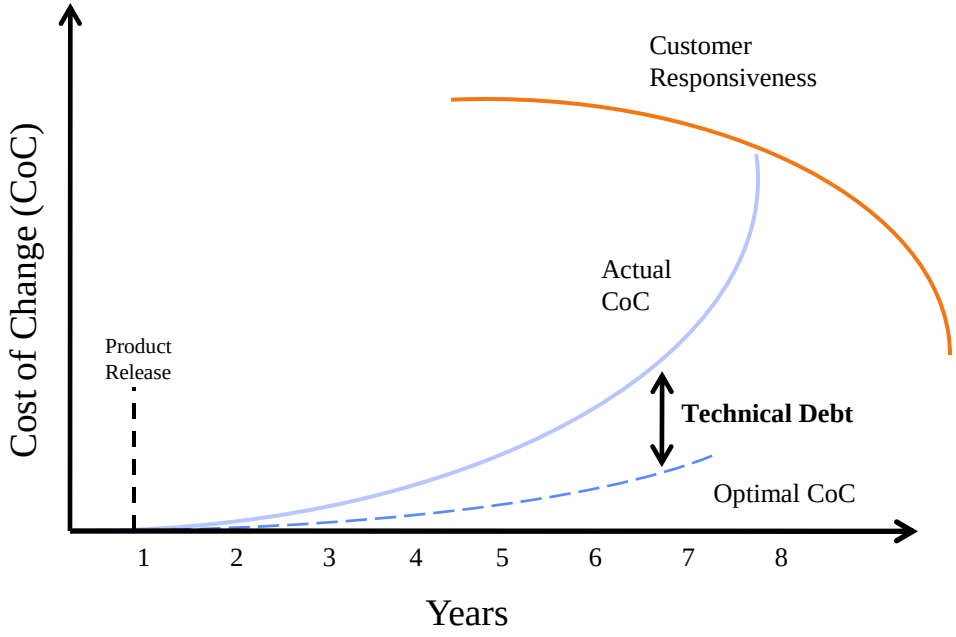
"Management of Technical Debt", Steve McConnell

Communicating about Technical Debt

- Shift from Technical vocabulary to a Financial vocabulary
- Use a projects Maintenance budget as a rough proxy for its technical debt service load
- Discuss Debt in terms of "money" instead of "features"
- Be sure you're taking the right kind of debt
- Treat the discussion of Debt as an ongoing dialog rather than a single discussion

Technical Debt Observations

"Agile Project Management", Jim Highsmith, second edition



Technical Debt Observations

"Agile Project Management", Jim Highsmith, second edition

- ***"When product development teams give lip service to technical excellence, when project and product managers push teams beyond quickness into hurrying, technical debt is incurred"(pp 216)***
- ***"Technical debt can arise during initial development, ongoing maintenance (keeping a product at its original state), or enhancements (adding functionality)."(pp 216)***

When does the insertion of Technical Debt have its greatest impact?

Technical Debt Observations

"Agile Project Management", Jim Highsmith, second edition

- ***"Without a firm dedication to long-term technical debt management, development groups are pressured into increasing technical debt trap. As the debt gets worst, the delays become greater. As the delays lengthen, the pressure increases, usually leading to another hurried implementation, which increases the technical debt yet again."(pp 217)***

Technical Debt Observations

"Agile Project Management", Jim Highsmith, second edition

- ***"It must be noted that managing technical debt does not keep products from becoming obsolete. A technical debt strategy does not attempt to stave off eventual obsolescence, but keeps the cost of change low so that customer responsiveness remains as high as possible during a product life."(pp 217)***

Technical Debt Observations

"Agile Project Management", Jim Highsmith, second edition

- ***"Holding the cost of change down by not changing only means that when change has to happen, neither the product nor the people will be ready for it."(pp 217)***

Types Debt

"Managing Software Debt: Building for Inevitable Change", Chris Sterling

- *Software Debt*
  - *Composed of the following forms of "Debt"*
    - *Technical Debt, Quality Debt, Configuration Management Debt, Design Debt & Platform Debt*
  - *Indicators of Software debt are the following:*
    - *Do you have "like-to-like" migration?*
    - *Do you have "limited expertise" available?*
    - *Do you have "expensive release stabilization" phases?*
    - *Do you have "increased cost for regression testing" your software assets?*

Types Debt

"Managing Software Debt: Building for Inevitable Change", Chris Sterling

- ***Technical Debt***
  - ***These are activities that a team or team members choose not to do well now and will impede future development if left undone***
- ***Quality Debt***
  - ***There is a diminishing ability to verify the functional and technical quality of software***
- ***Configuration Management Debt***
  - ***Integration and release management becomes more risky, complex and error-prone***
- ***Design Debt***
  - ***The cost of adding features is increasing toward the point where it is more than the cost of writing from scratch.***
- ***Platform Debt***
  - ***The availability of people to work on software changes is becoming limited or cost-prohibitive.***

At Systems level, each of these items constitute decisions which will cause Systems Engineering Technical Debt to occur.

Types Debt

"Managing Software Debt: Building for Inevitable Change", Chris Sterling

- *Technical Debt*
  - *"..the decay of component and inter-component behavior when the application functionality meets a minimum standard of satisfaction for its users"*
  - *Produced by work patterns:*
    - *Schedule pressure*
      - *"Excessive pressure causes team to take short cuts to meet expectation of management and customer"*
    - *Duplication*
      - *"..because of cut-and-paste tactics results in teams making even simple changes in more than one place."*
    - *Mentality of getting it right the first time*
      - *"..incorrect assumptions about what we can know about the future."*
  - *"..payoff Technical Debt immediately, insert strategically placed runtime exceptions, and add technical debt to the Product Backlog."*

Types Debt

"Managing Software Debt: Building for Inevitable Change", Chris Sterling

- *Quality Debt*
  - *"To sustain the internal quality of software, teams must approach development in a disciplined way. Common approaches that teams use to sustain internal quality are the following:*
    - *Sustainable pace*
    - *Early identification of internal quality problems*
    - *Close collaboration*
    - *Refactoring*
    - *Small batches of work*
    - *Defining technically done*
    - *Potentially shippable product increments*
    - *Single work queue"*

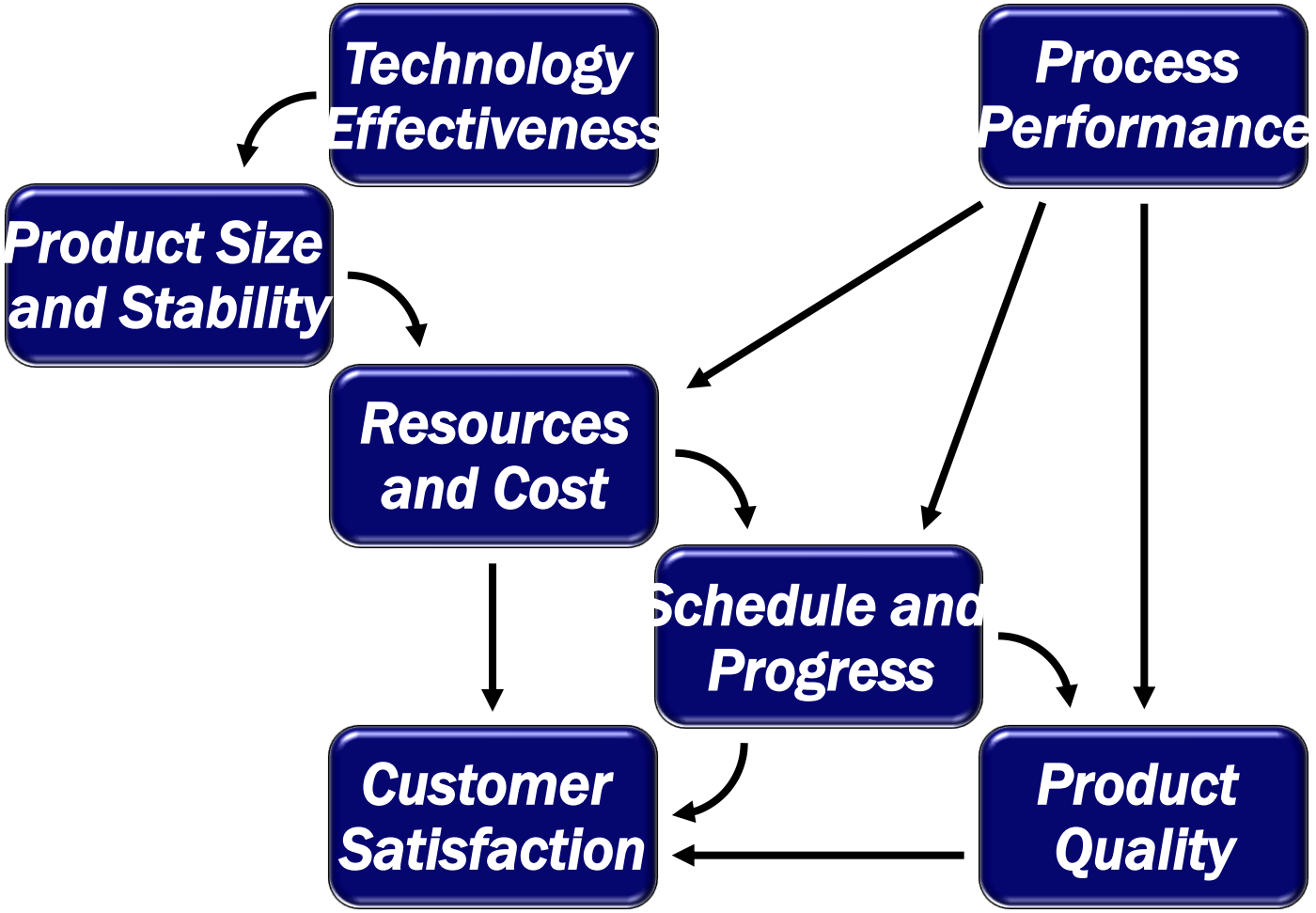
What Is A Measure?

- ***Measure***
  A variable to which a value is assigned to represent one or more attributes of an entity

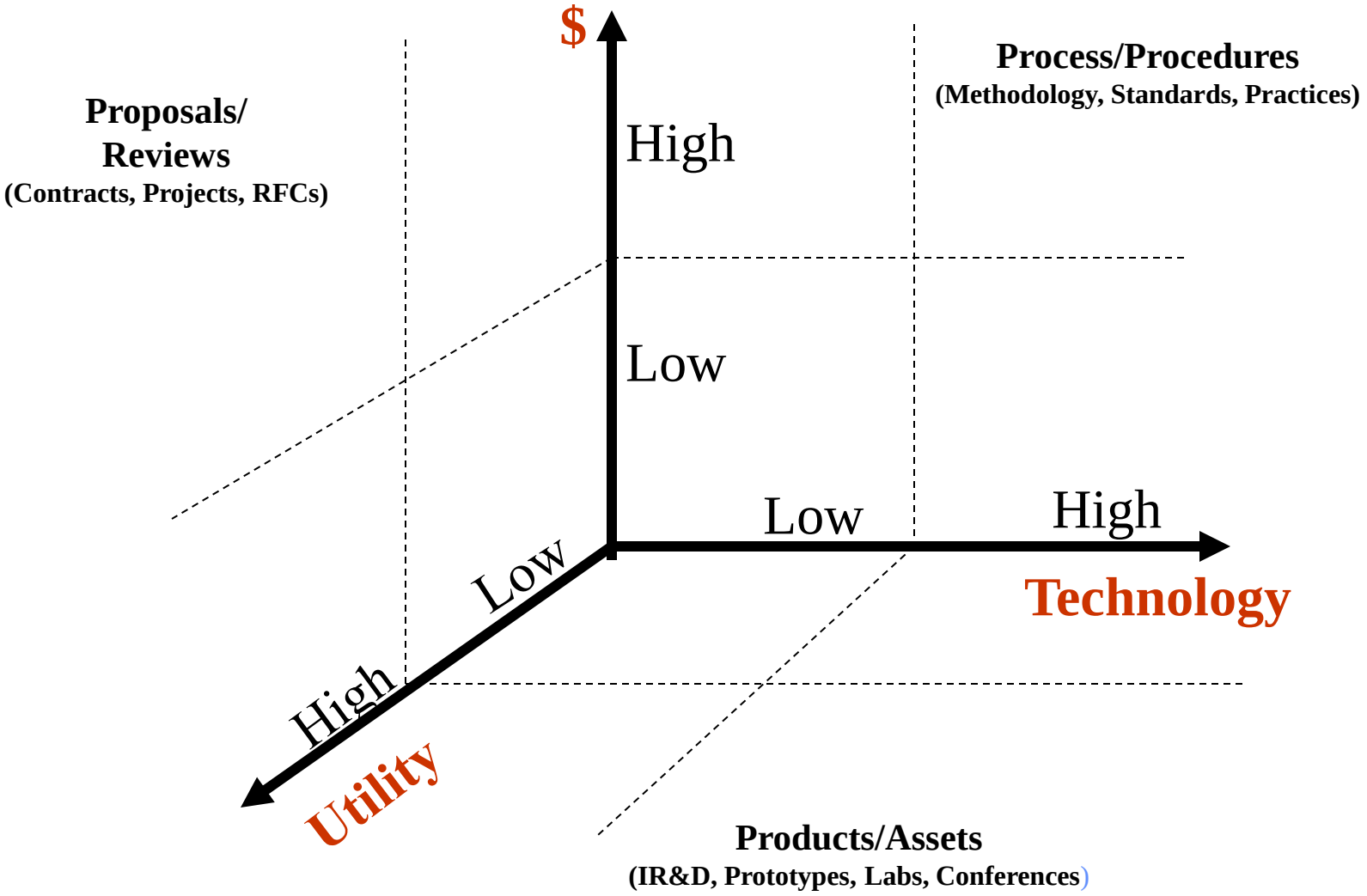
- ***Measures***
  Used collectively to refer to base measures, derived measures, and indicators

Source: "Introduction to PSM Course, Version 7.0", July 30,2012, Portsmouth, VA

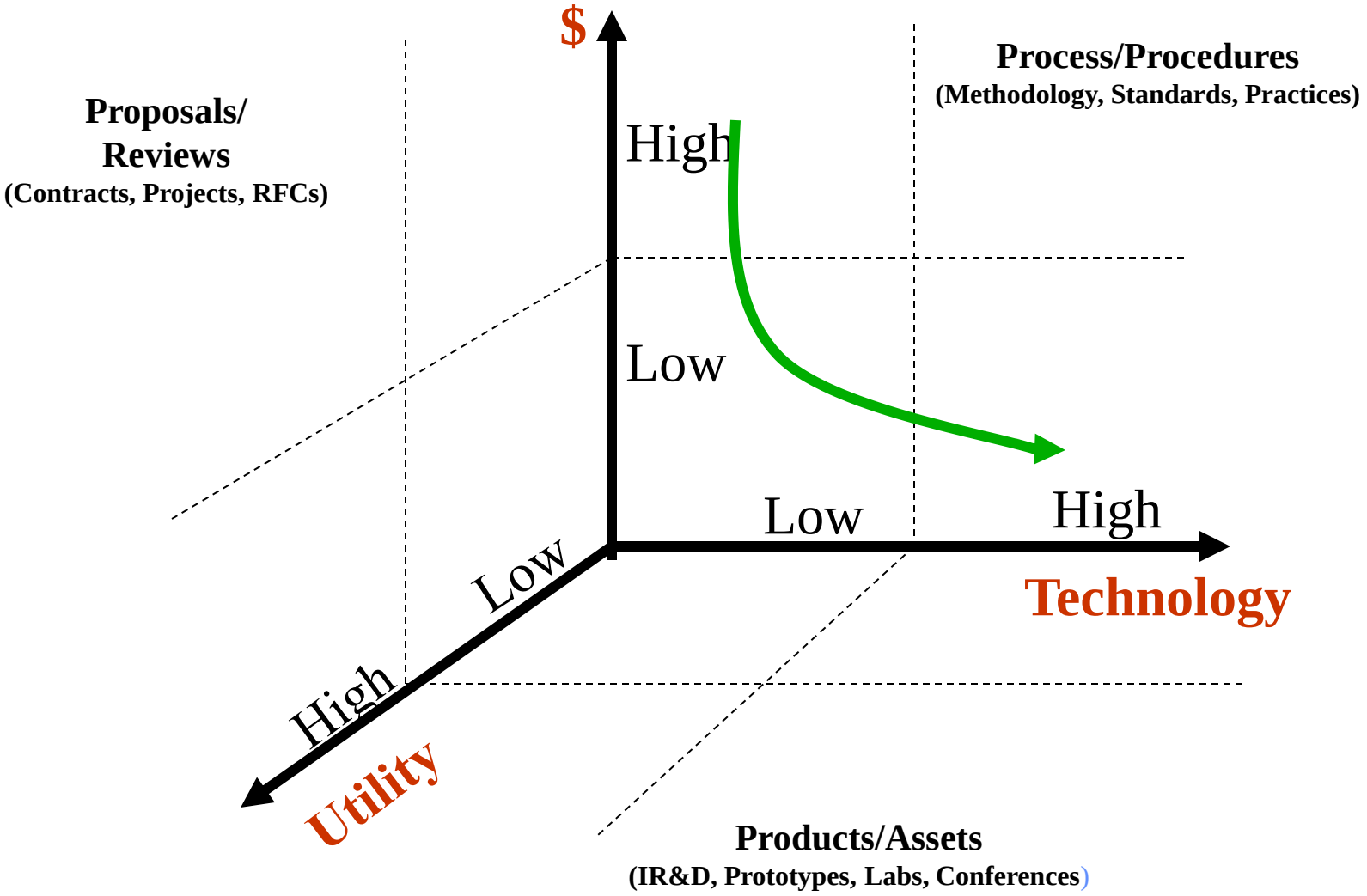
Integrated Analysis Model



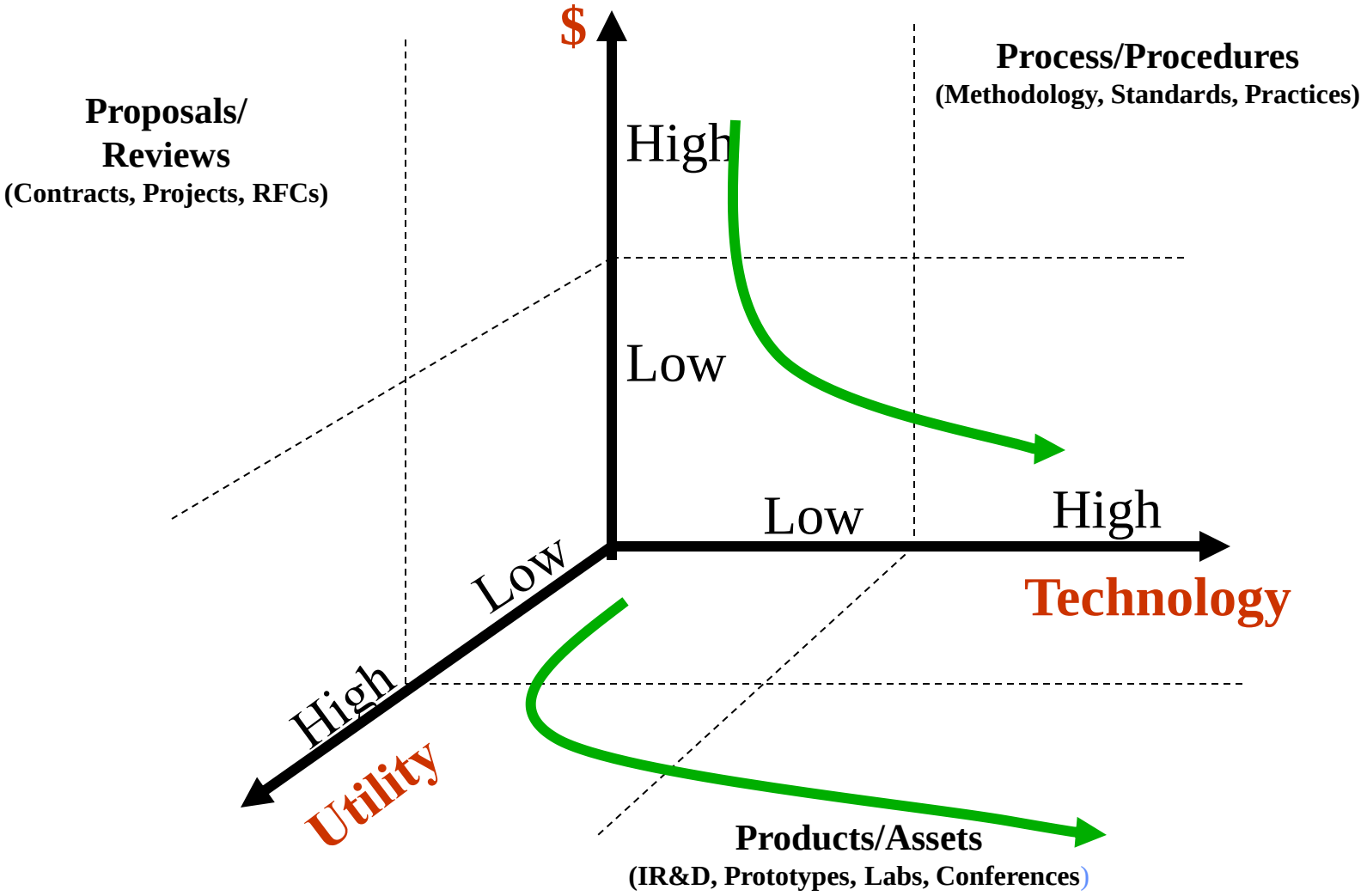
"The Challenge"



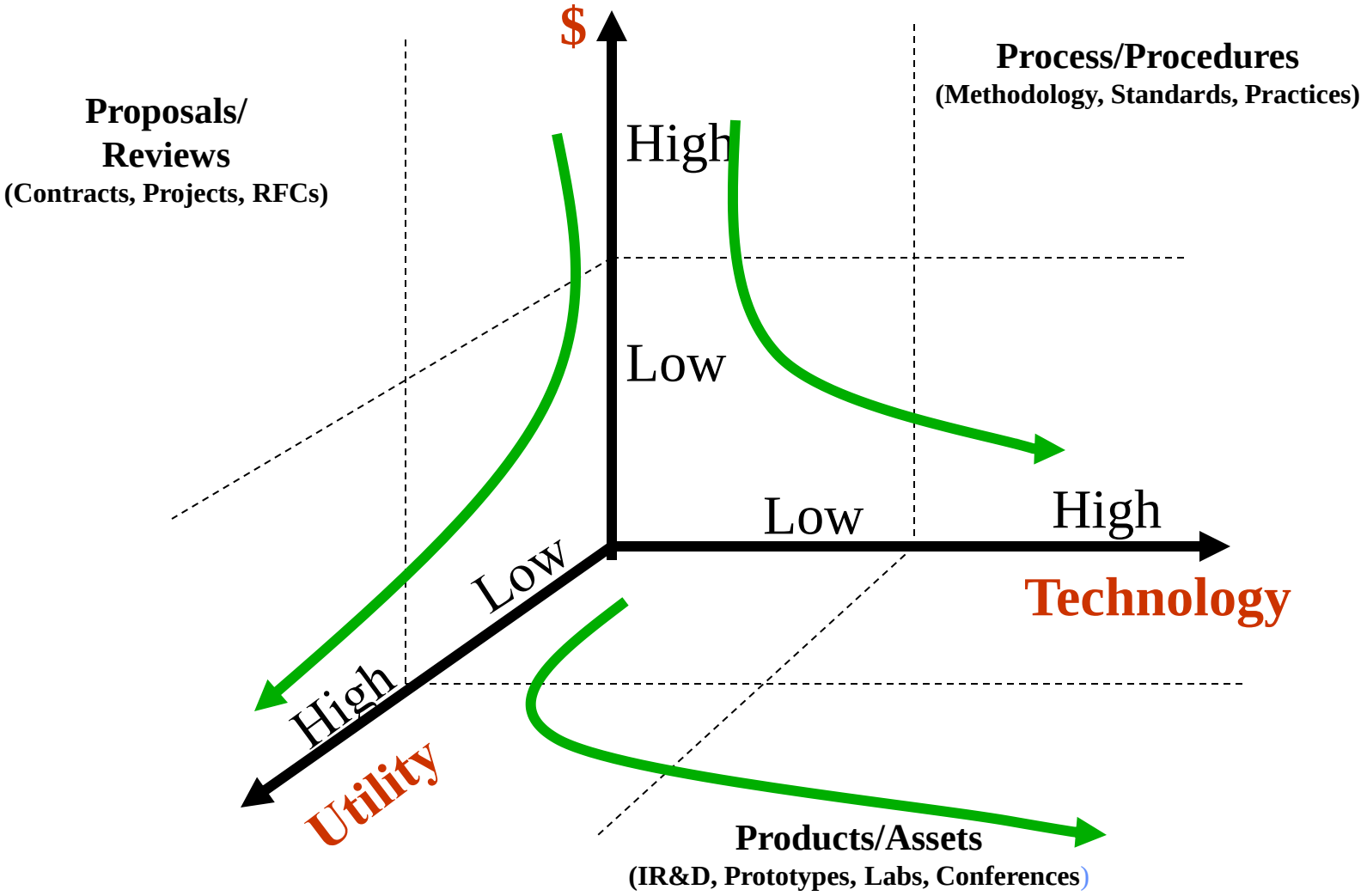
"The Challenge"



"The Challenge"

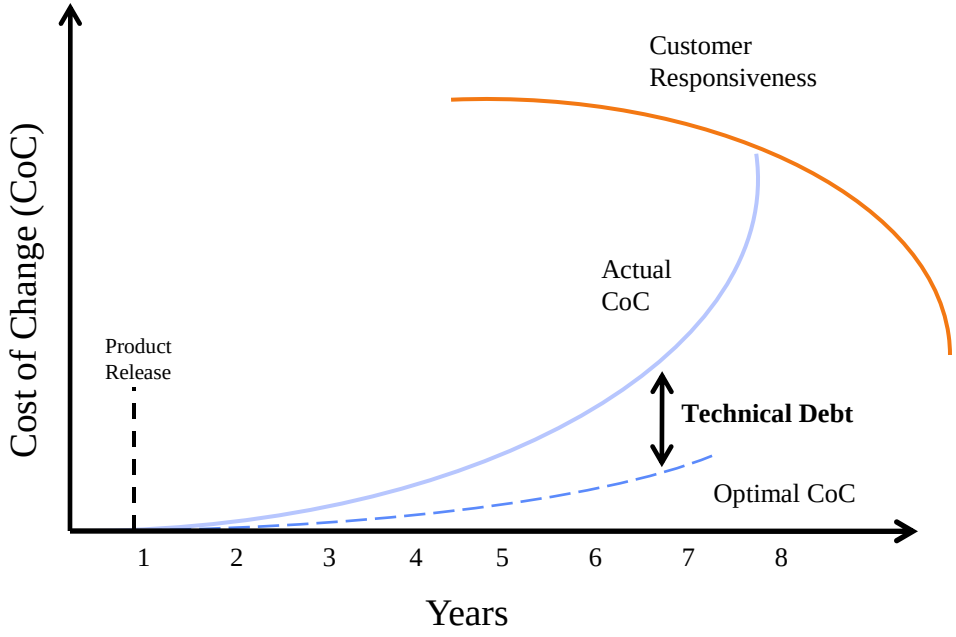


"The Challenge"



Technical Debt Observations

"Agile Project Management", Jim Highsmith, second edition



"Managing Technical Debt helps ensure reliable delivery *today* and ready adaptation to *tomorrow's* customer needs."
"*Rising technical debt is the single largest impediment to continuing product viability*"

Jim Highsmith, "Agile Project Management", second edition

When constrained, Project Managers make trade-off decisions that impact, in order:

1. Development progress
2. Product technical performance
3. Product quality and rework
4. System usability
5. Cost

Reference: Results from TAI systemic analysis of 50 large-scale DoD projects.

Source: "Introduction to PSM Course, Version 7.0", July 30,2012, Portsmouth, VA

Types of Measures

Characterization Measures

- Quantify the attributes of an enterprise, organization, product, project, etc.
- Helps to describe and categorize entities based on their characteristics (size, weight, color, quality, frequencies of occurrence, etc.)
- Provide a general context

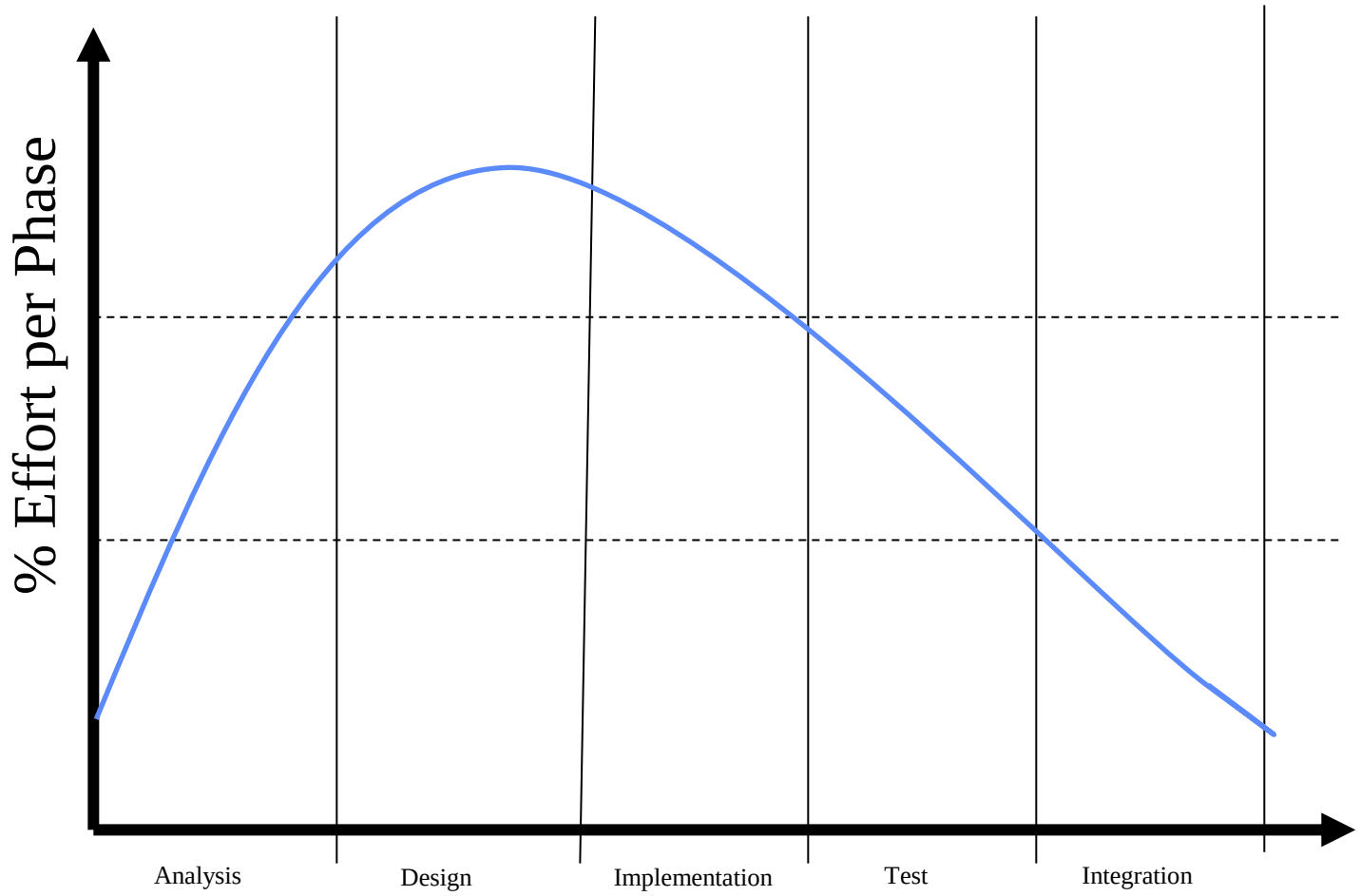
Activity Measures

- Tell you if you are doing what you said you were going to do
- Largely relate to schedules and planned accomplishments
- Are what most measurement processes initially focus on
- Usually structured as expected vs. actual values

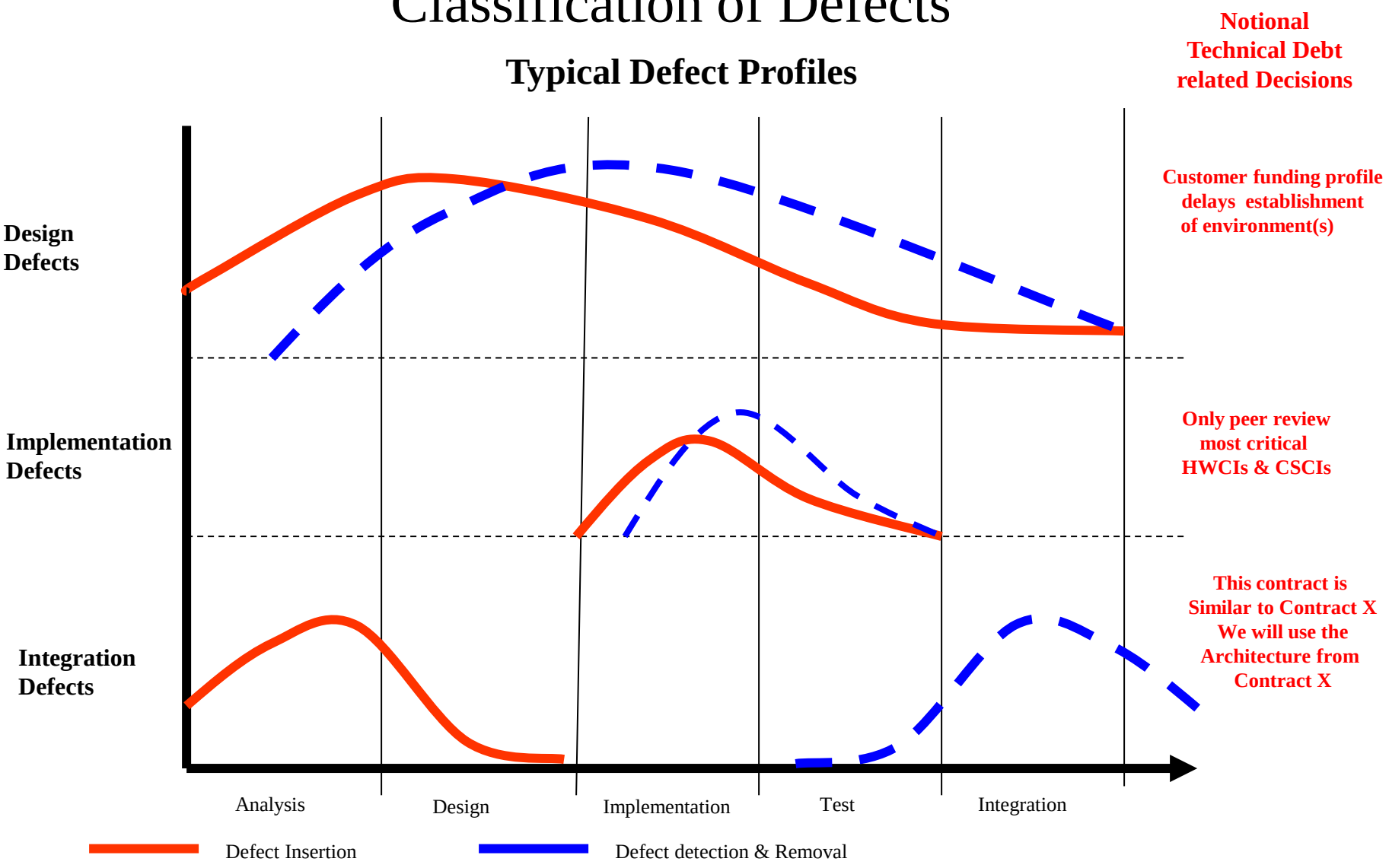
Workshop Exercise # 1
Identifying Technical Debt

- Development profile (Perfect World)
- Defect Density profile
- Development profile(Real World)
- Mapping Development Profiles

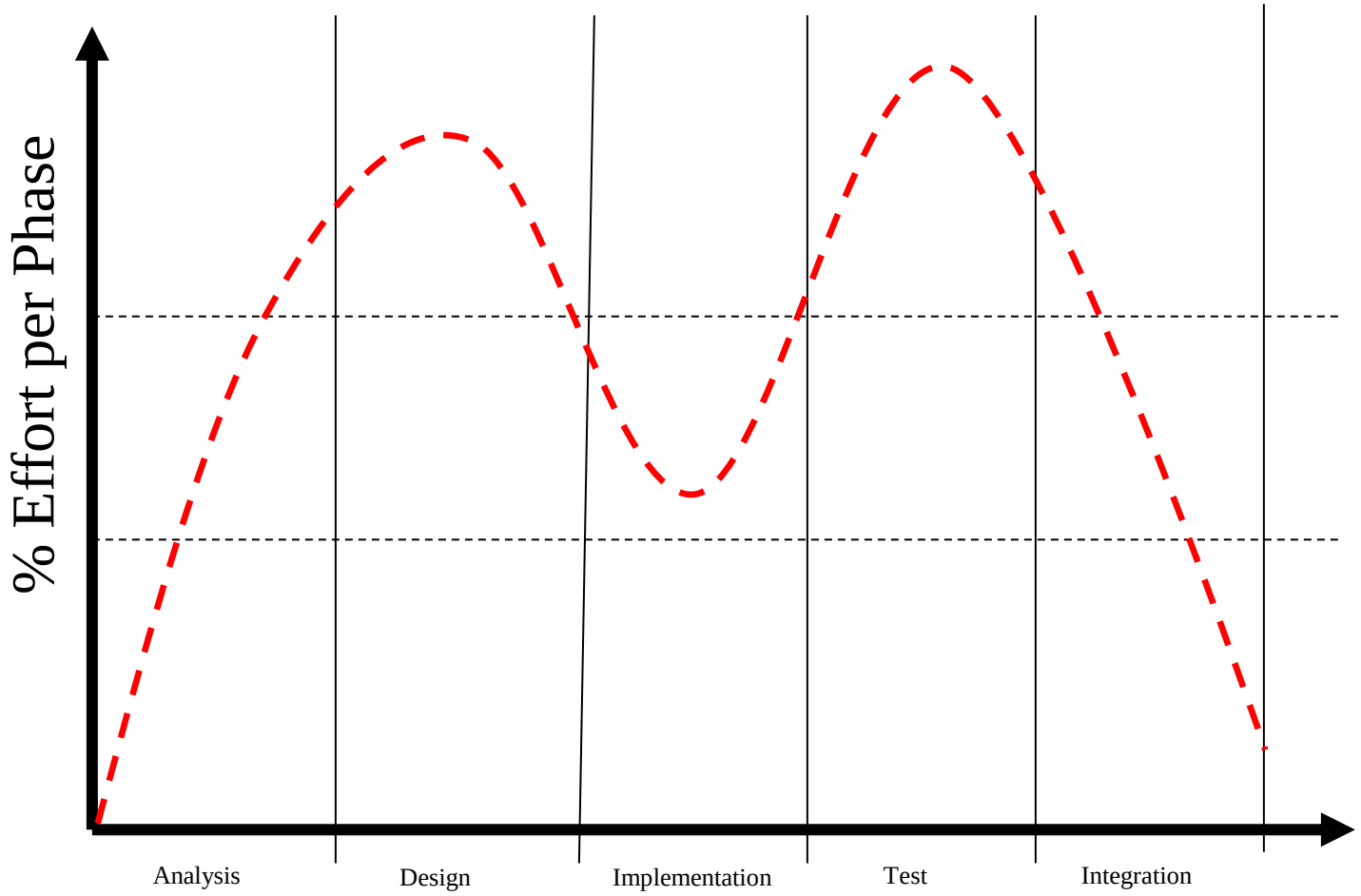
Development Cost(Perfect World)



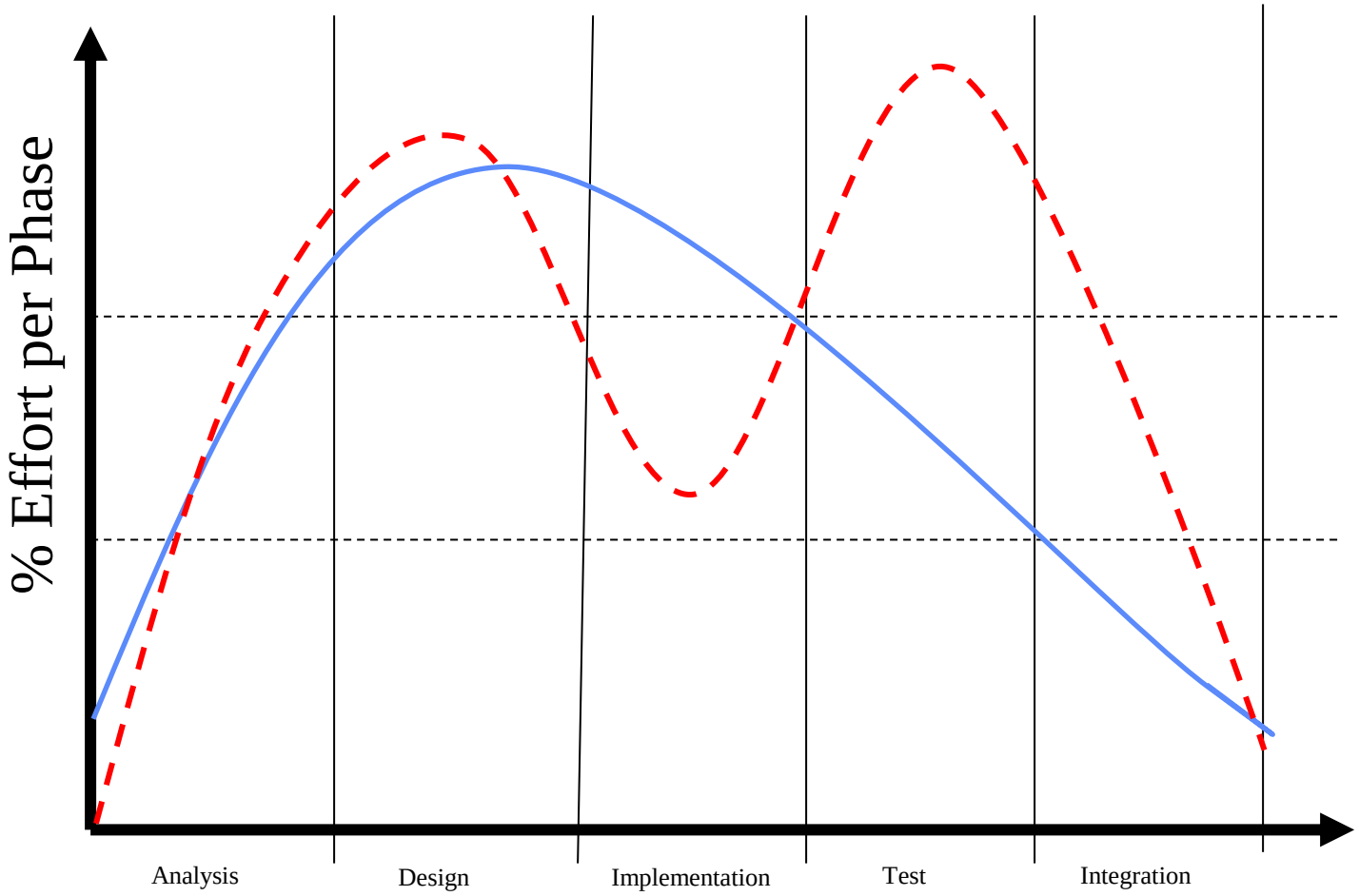
Classification of Defects



Development Cost(Real World)

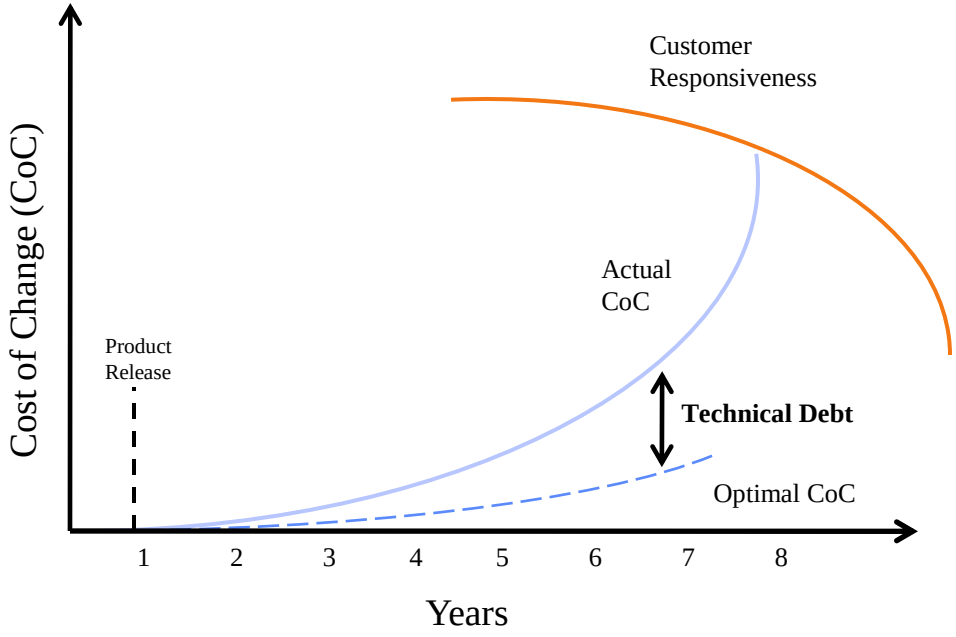


Development Cost

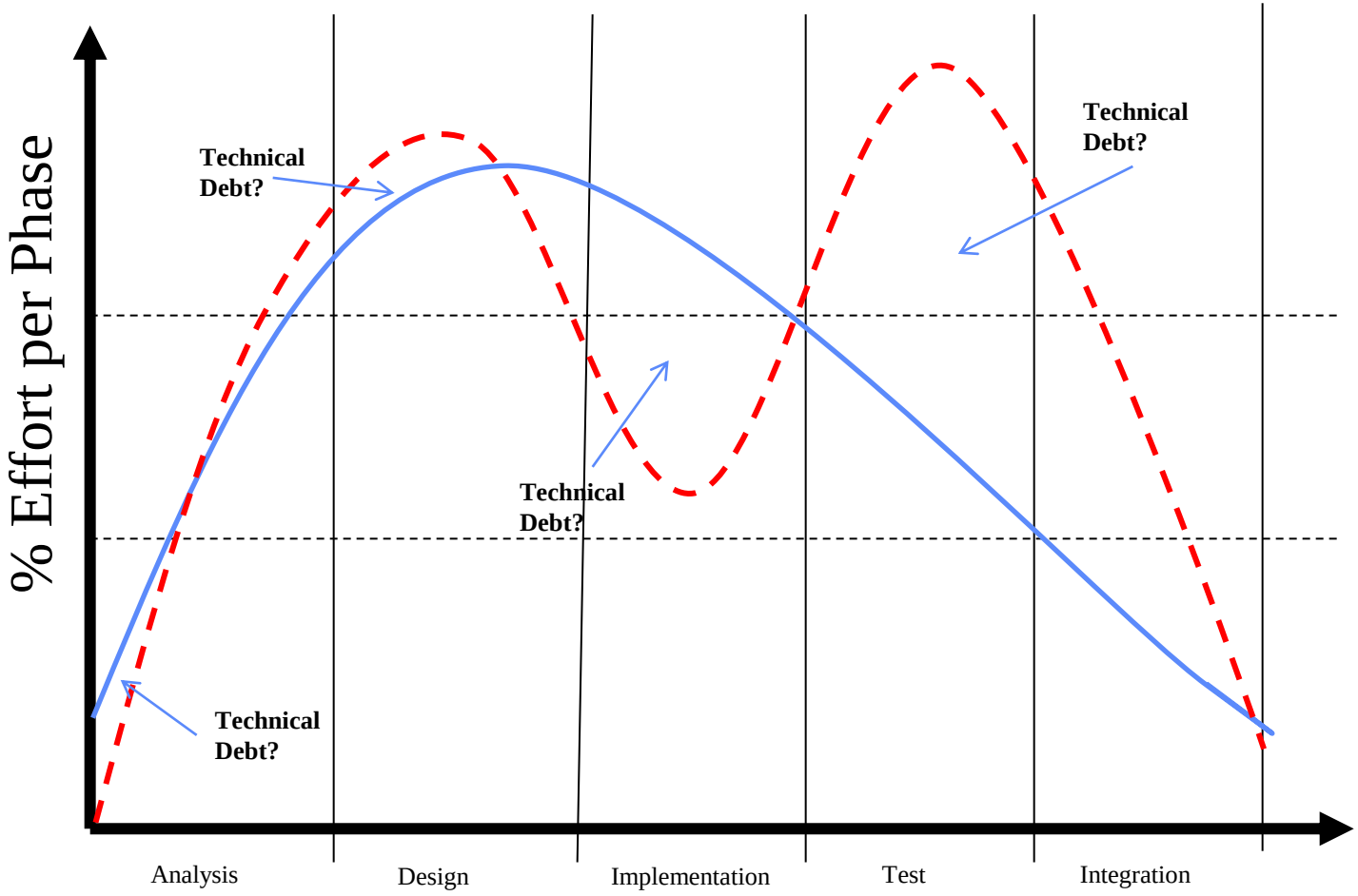


Technical Debt Observations

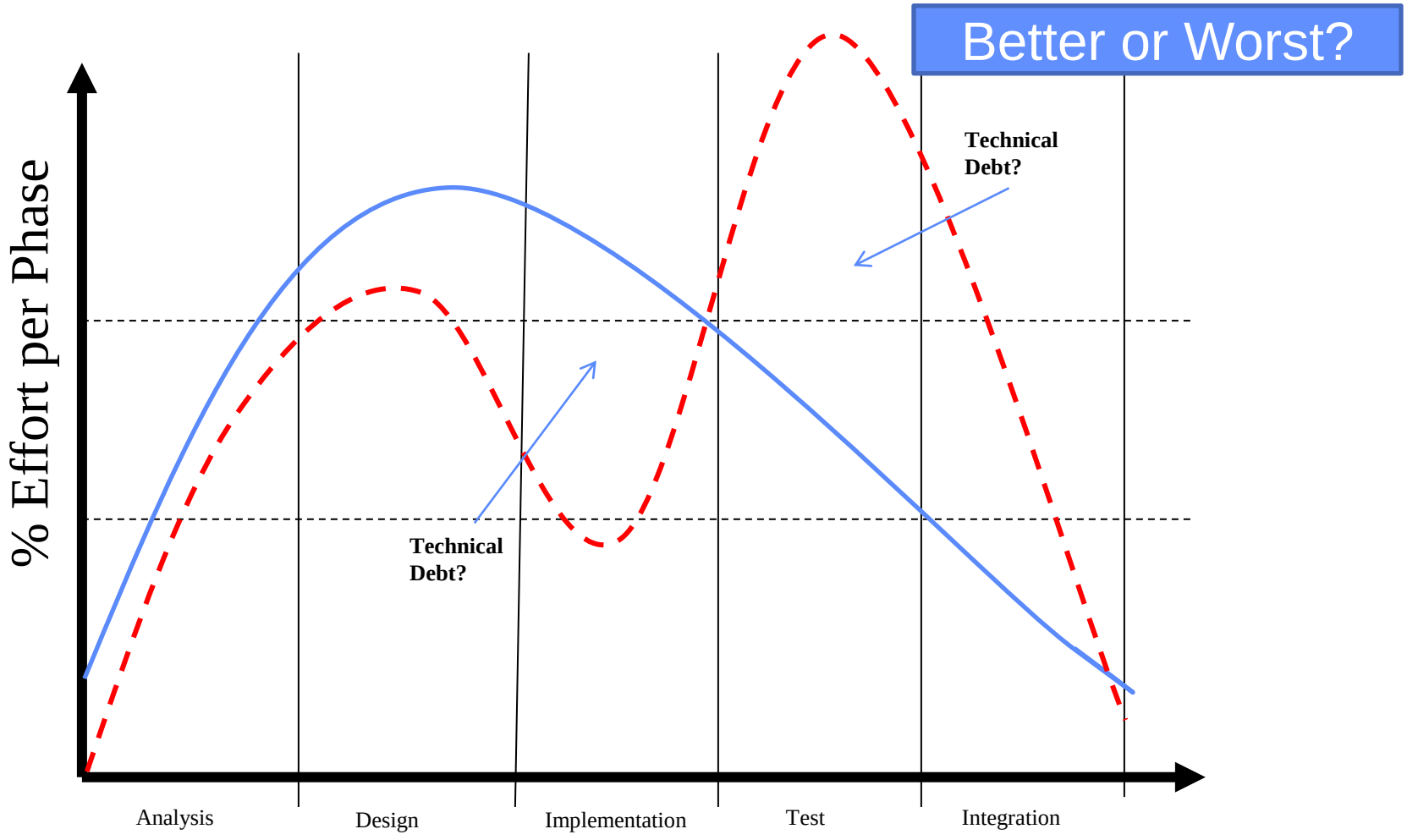
"Agile Project Management", Jim Highsmith, second edition



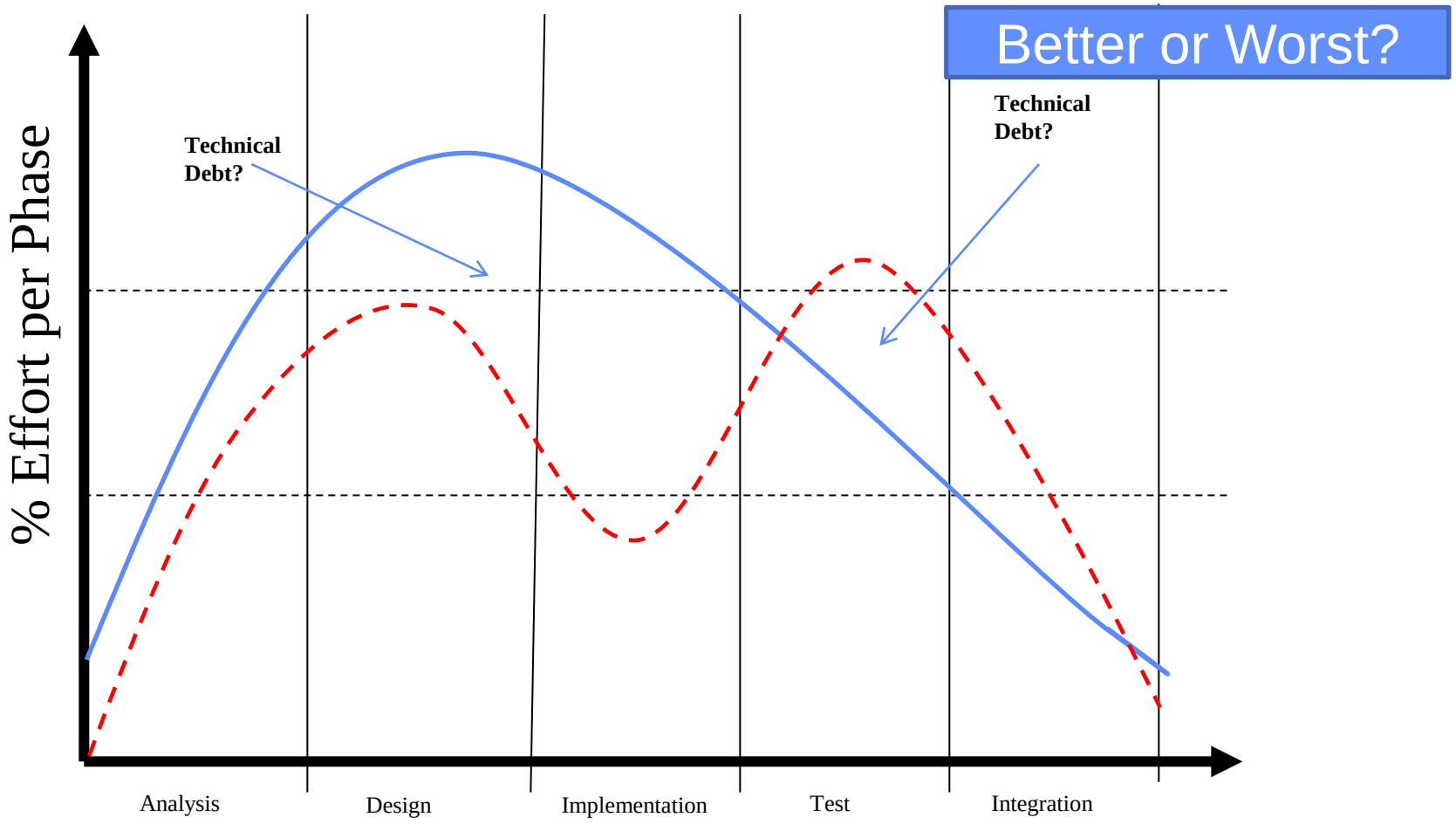
Development Cost



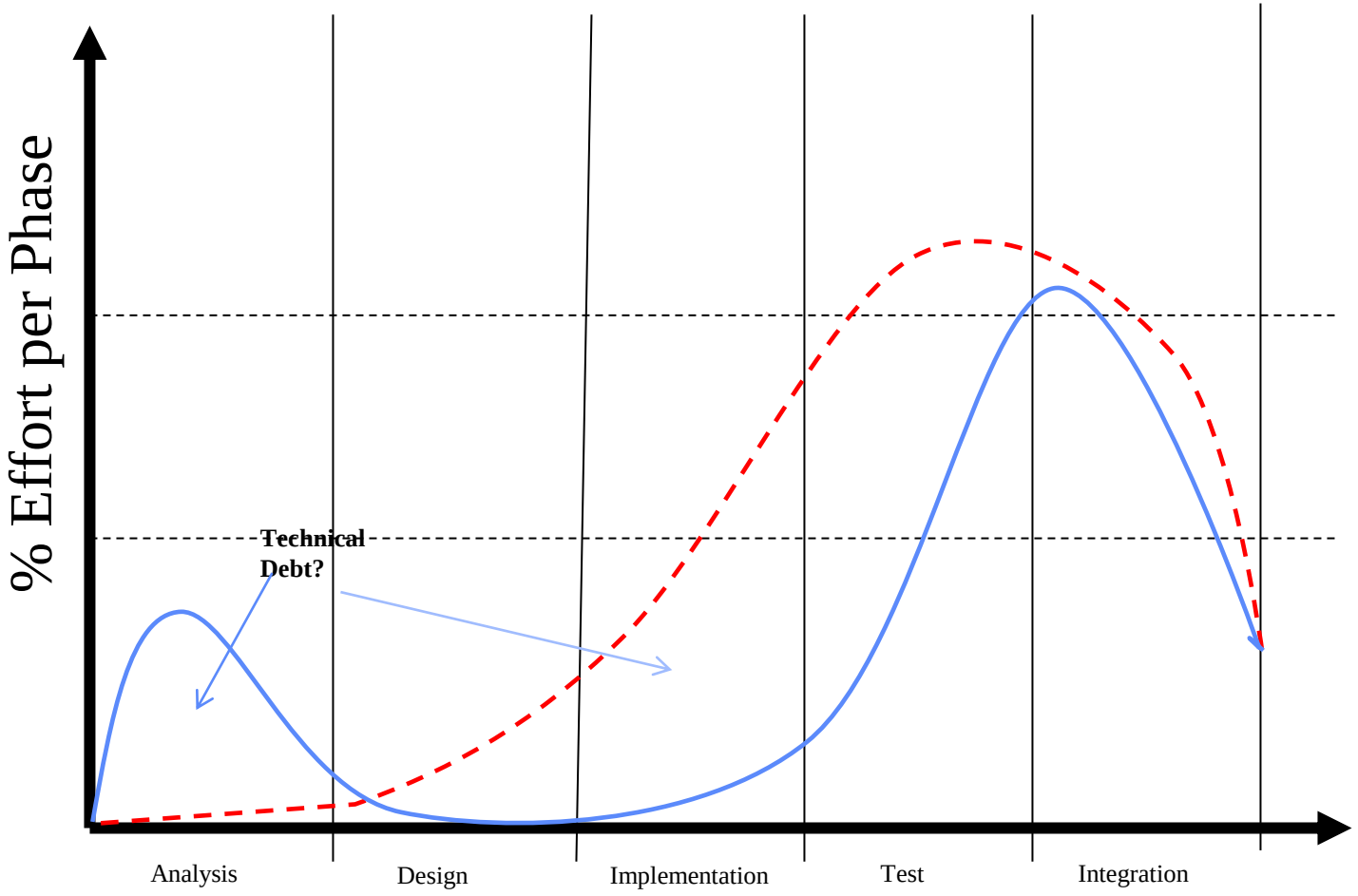
Development Cost



Development Cost



COTS Integration



Technical Debt

"Enabling Agility by Strategically Managing Architectural Technical Debt", Ipek Ozkaya

- ***"Practices intended to speed up the delivery of value to users, however, often result in high rework costs that ultimately offset the benefits of faster delivery, especially when good engineering practices are forgotten along the way. The rework and degrading quality often is referred to as technical debt"***
- ***"For example, through our work on architecture-centric engineering, we often encounter projects that defer modifiability requirements, specifically portability."***

Technical Debt

"Enabling Agility by Strategically Managing Architectural Technical Debt", Ipek Ozkaya

- *"Our current work focuses on architectural technical debt, which involves architectural decisions made to defer necessary work during the planning or execution of software projects, such as short-cuts taken in designing the structure of the system that may require rework."*
- *"We are particularly interested in identifying the measureable aspects of architectural technical debt by exploring dependency analysis"*

Technical Debt

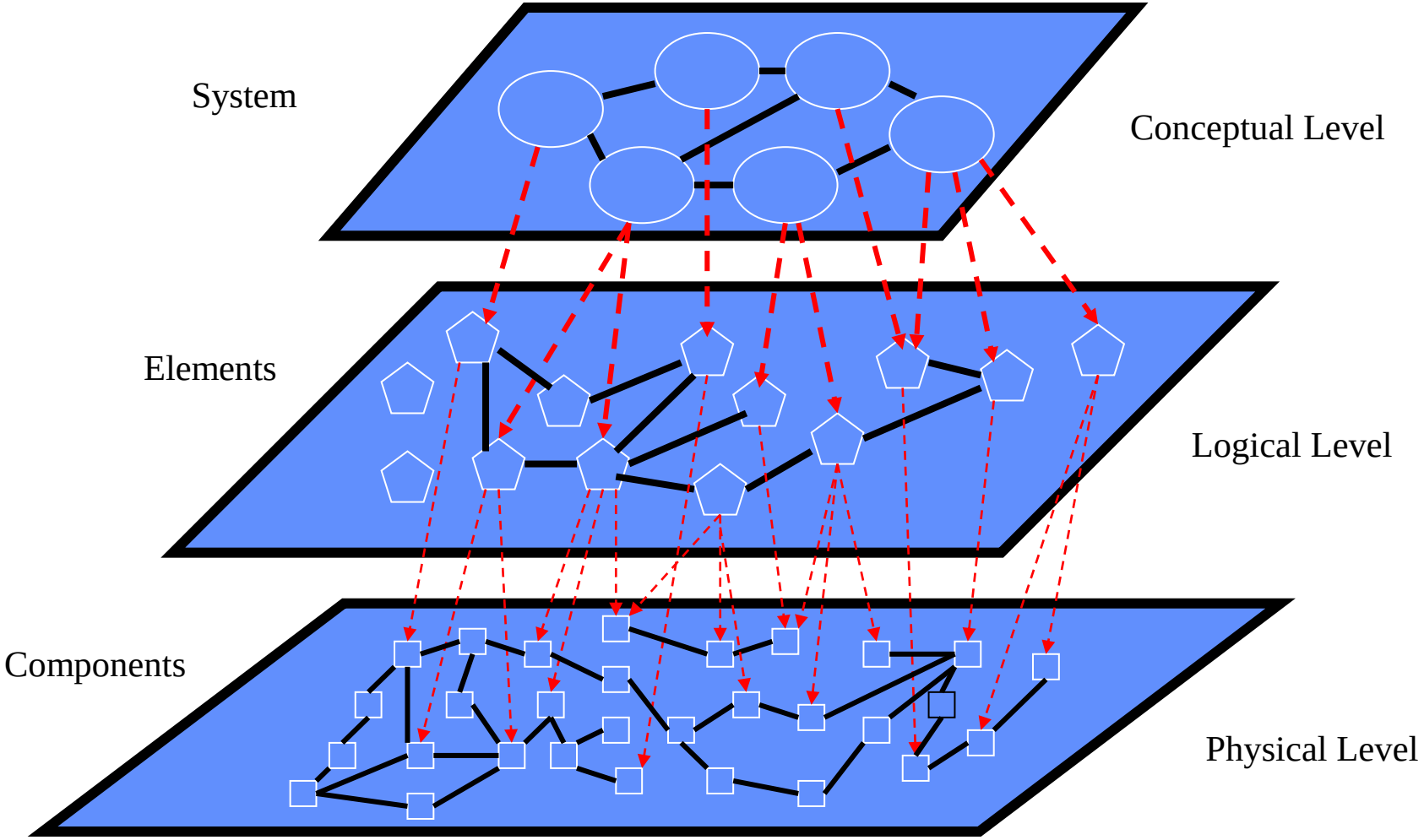
"Enabling Agility by Strategically Managing Architectural Technical Debt", Ipek Ozkaya

- ***"By the end of this project, we will produce a model for managing technical debt that will allow the incurrence of some debt to increase delivery tempo when needed, but prevent too much accumulation, which would impede the ability to deliver."***

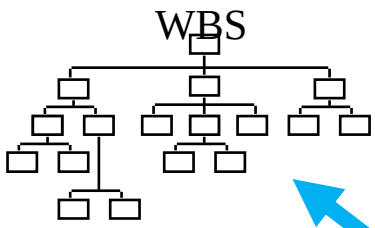
Reference: "Enabling Agility through Architecture",
Nanette Brown, Robert Nord, Ipek Ozkaya; CrossTalk-Nov/Dec 2010

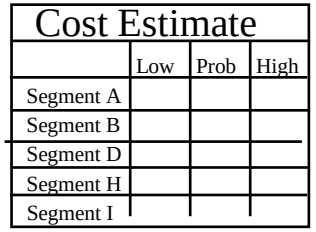
Workshop Exercise #2
Architecture & Technical Debt

Levels of Architecture (Conceptual to Logical to Physical Mapping)

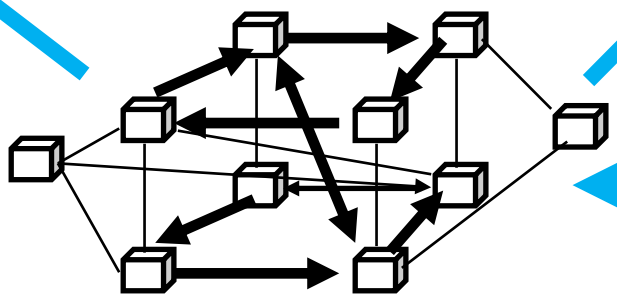


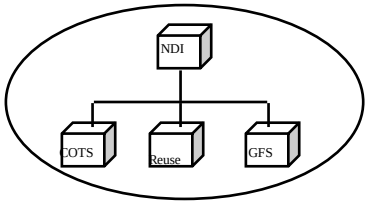
Notional Architecture Process

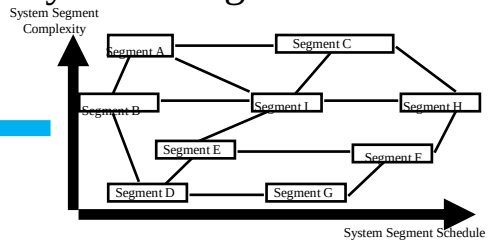




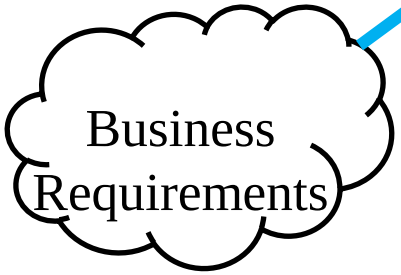
Architecture

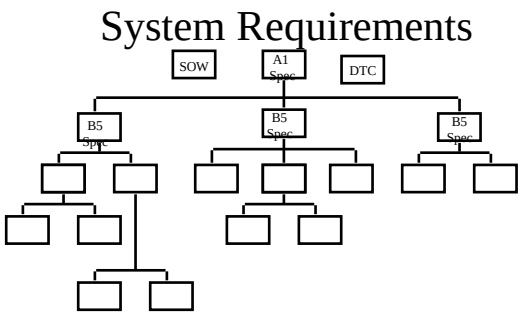


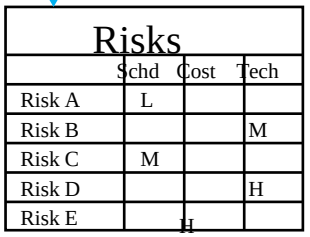




Where can Technical Debt be incurred in this process?

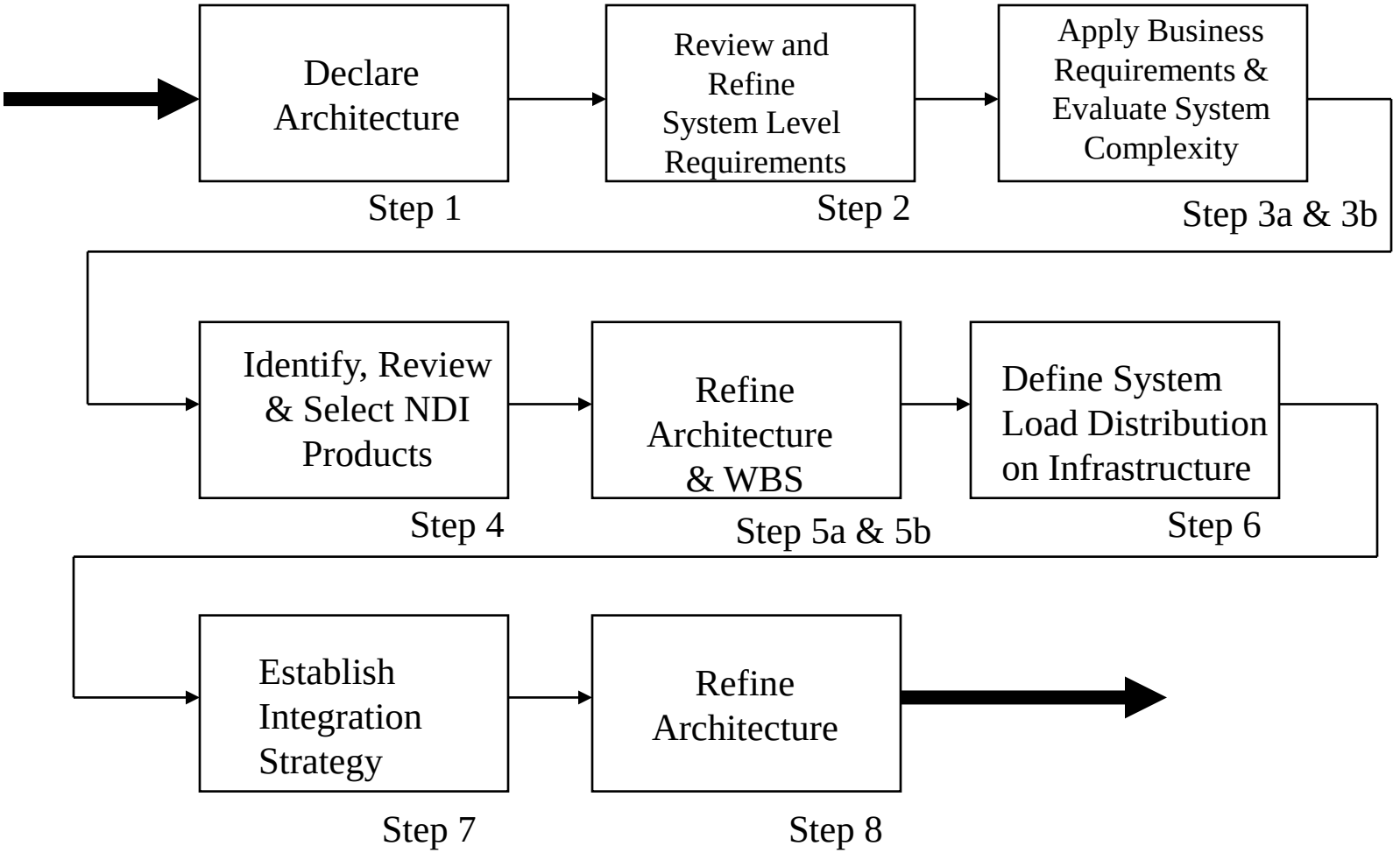




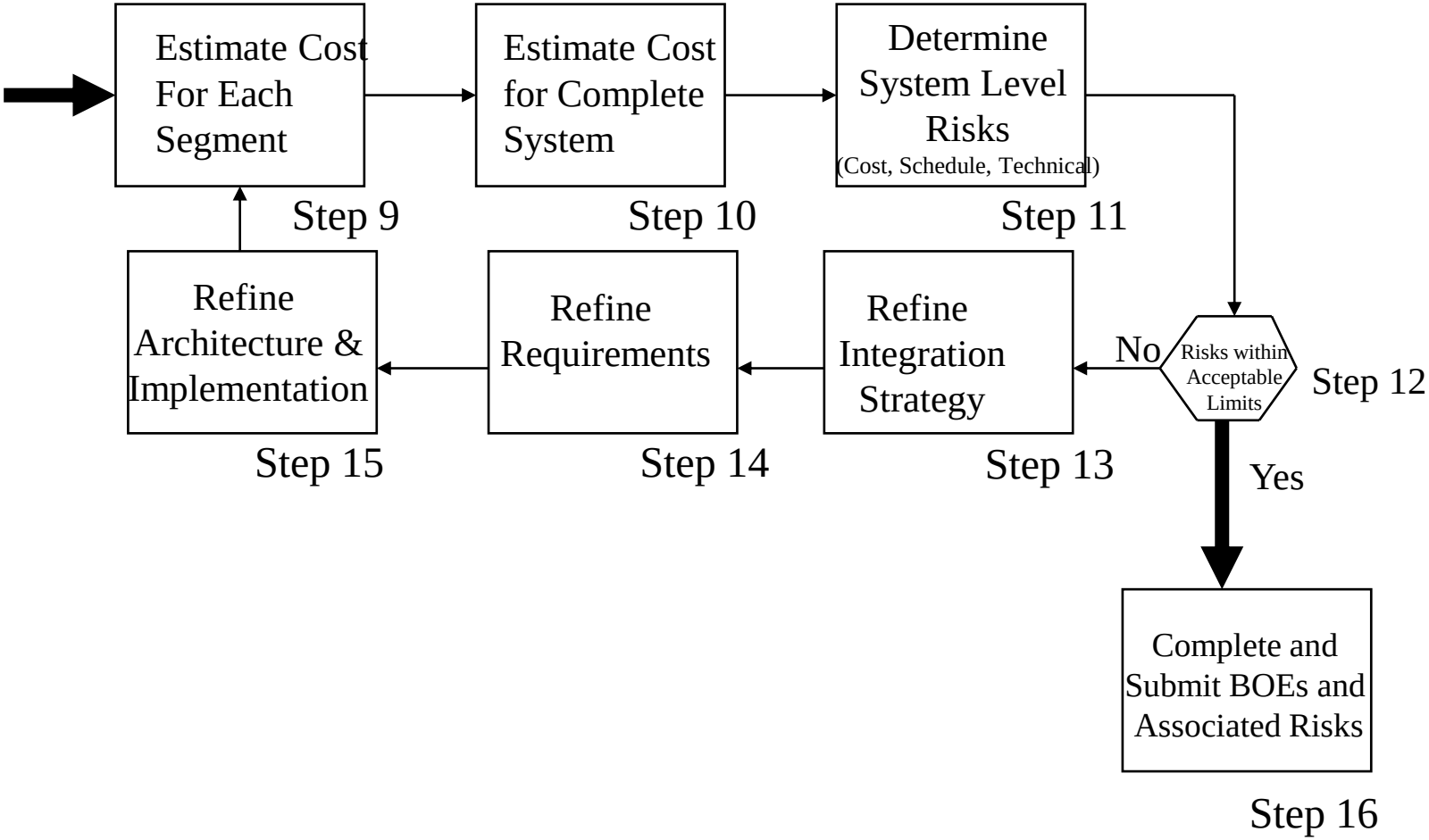


(Domain Knowledge)

Notional Architecture Process

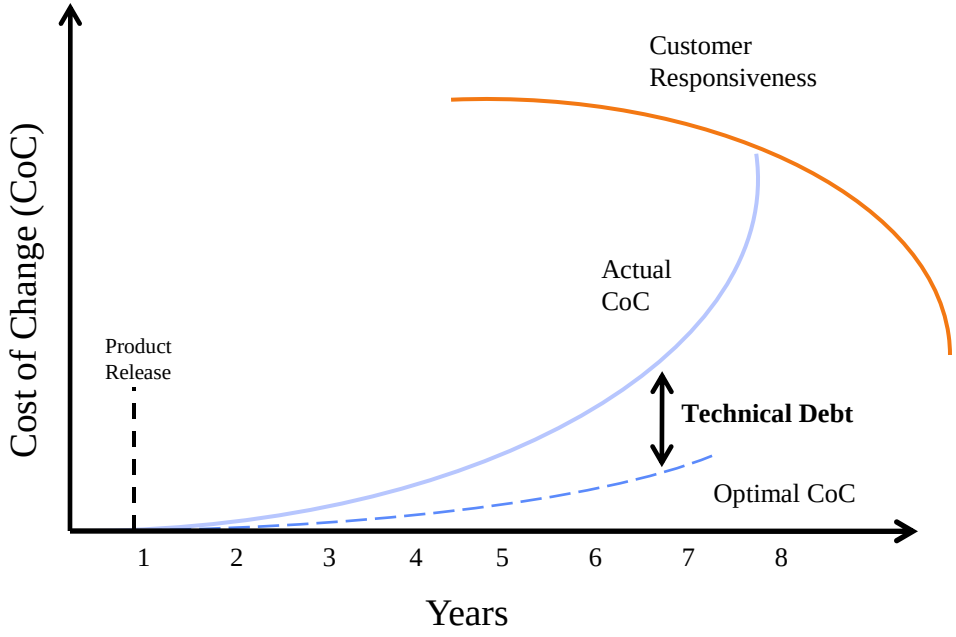


Notional Architecture Process (continuation)

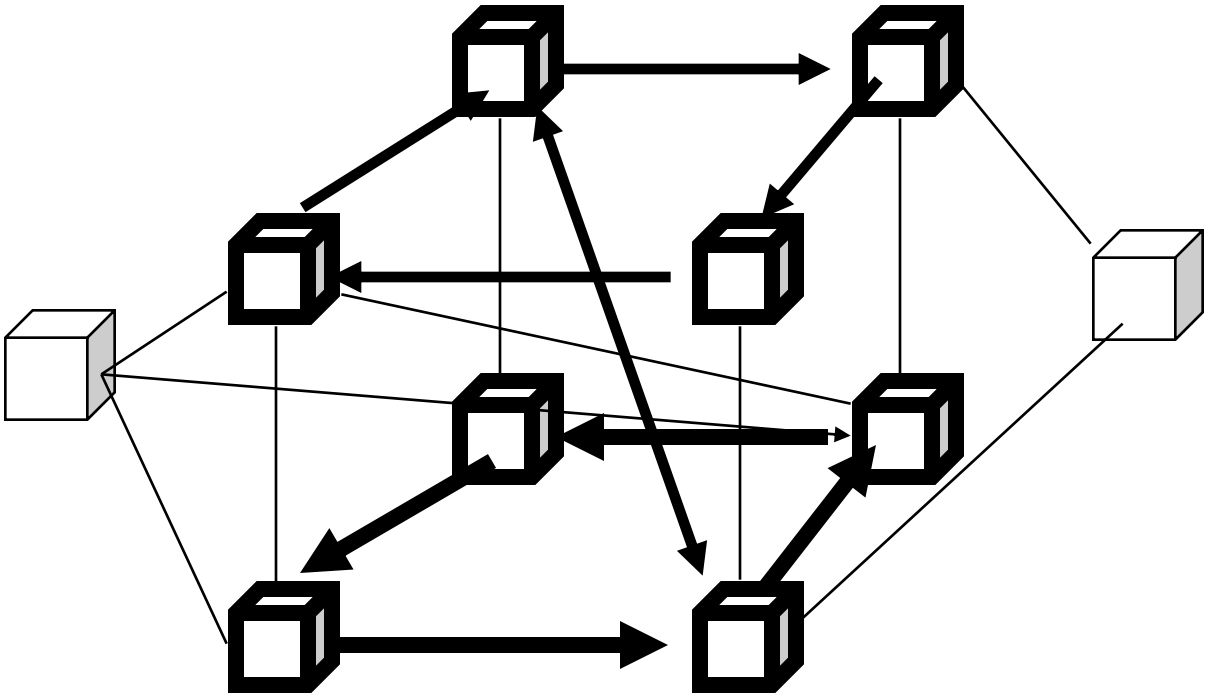


Technical Debt Observations

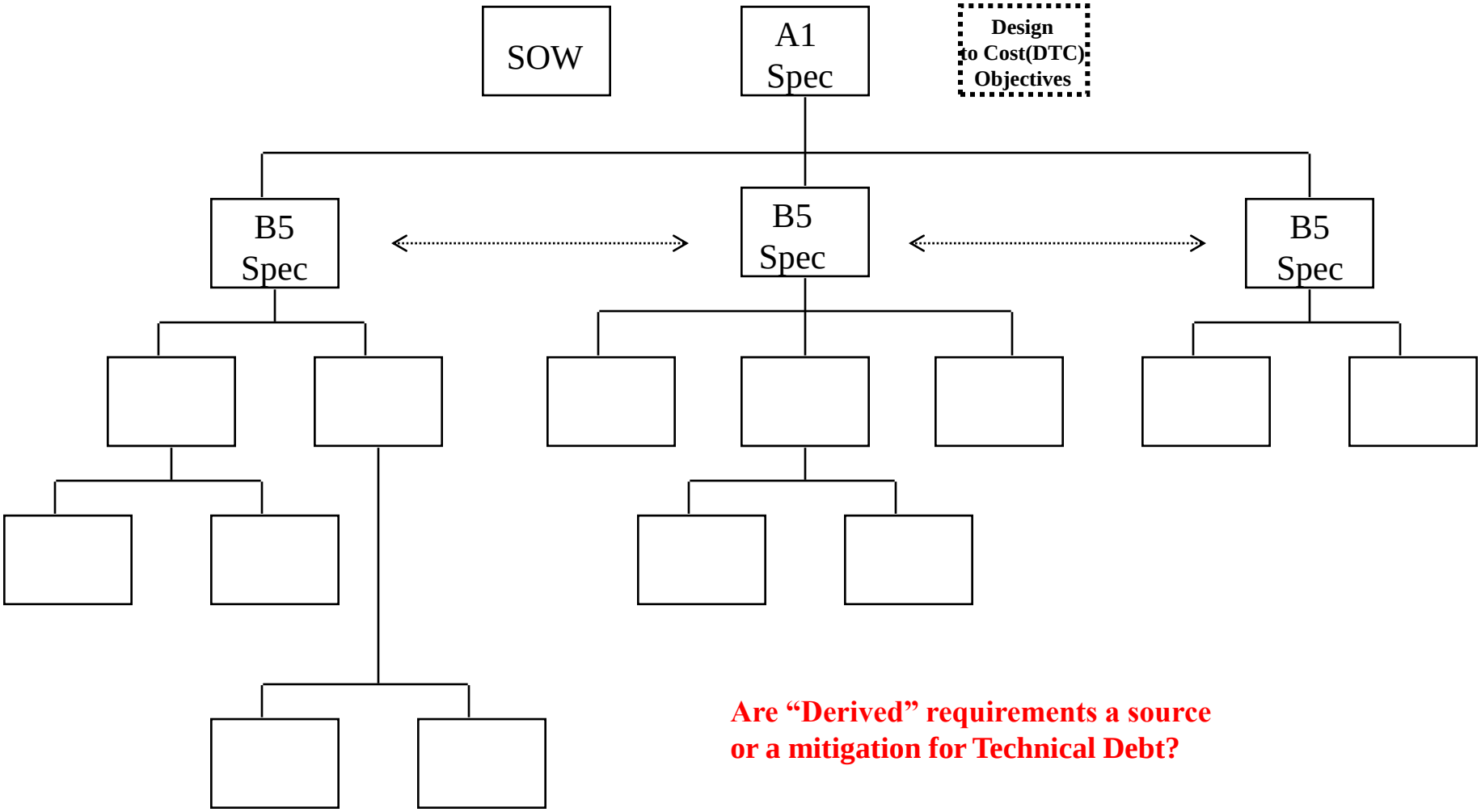
"Agile Project Management", Jim Highsmith, second edition



Step 1 Declare Architecture

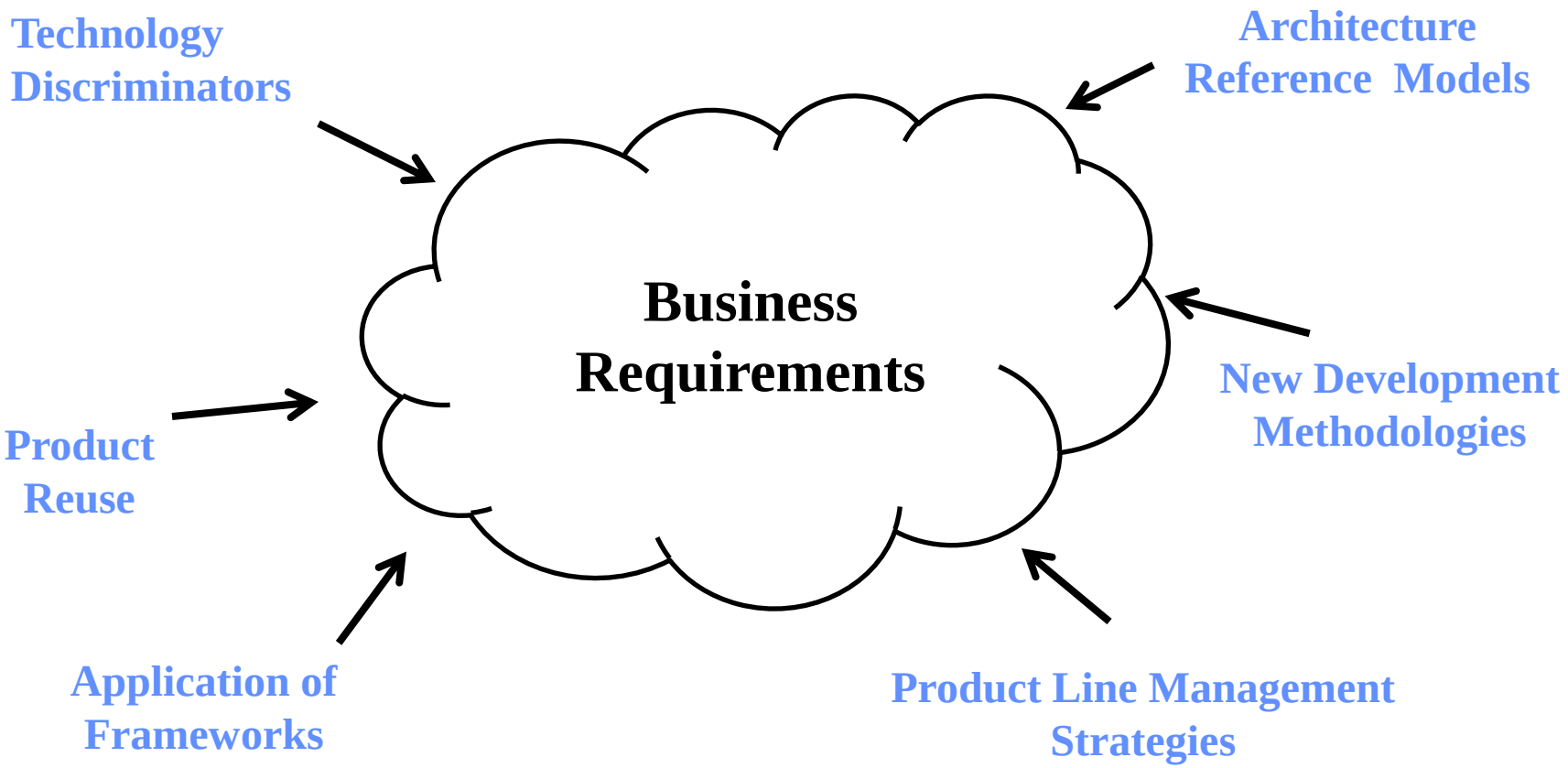


Steps 2 & 14 Review & Refine System Requirements



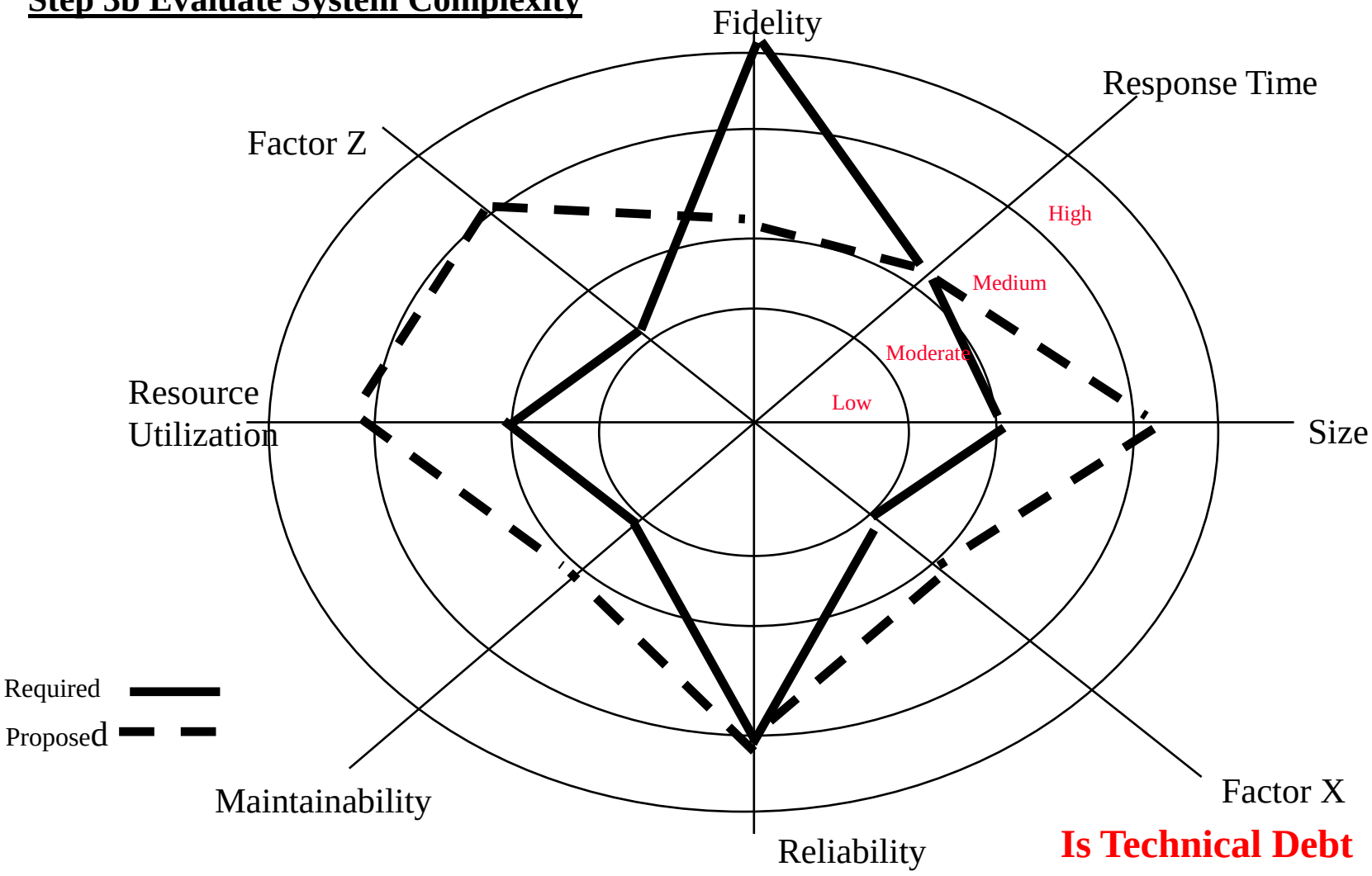
Are "Derived" requirements a source or a mitigation for Technical Debt?

Step 3a Business Requirements



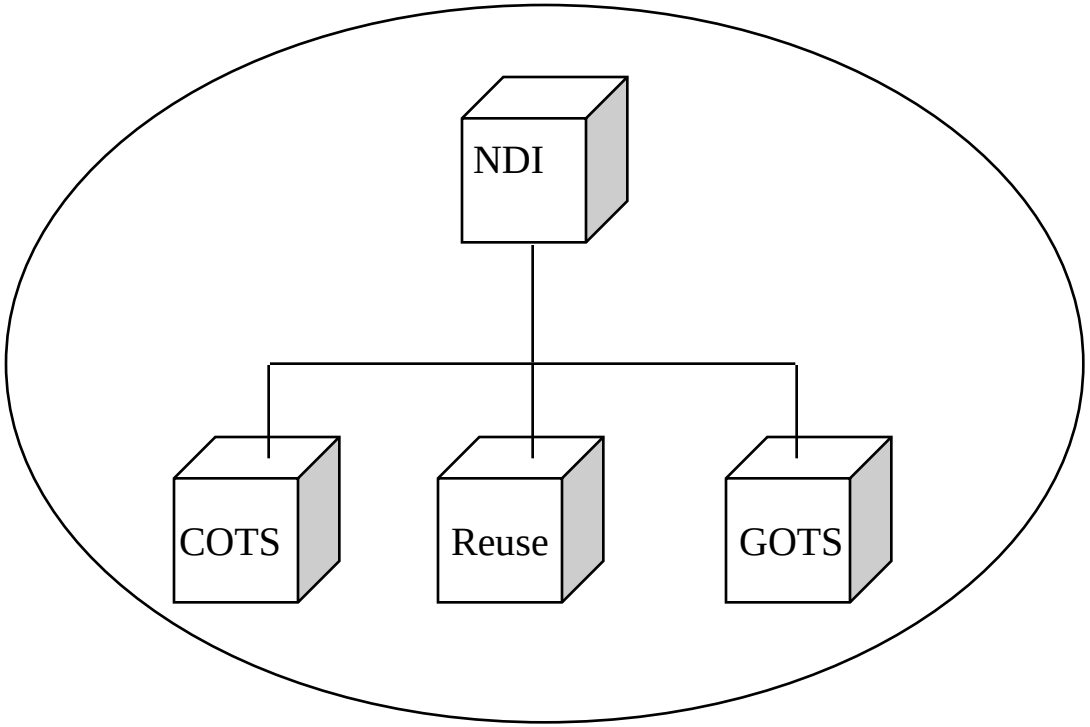
Is this a Source of Technical Debt?

Step 3b Evaluate System Complexity



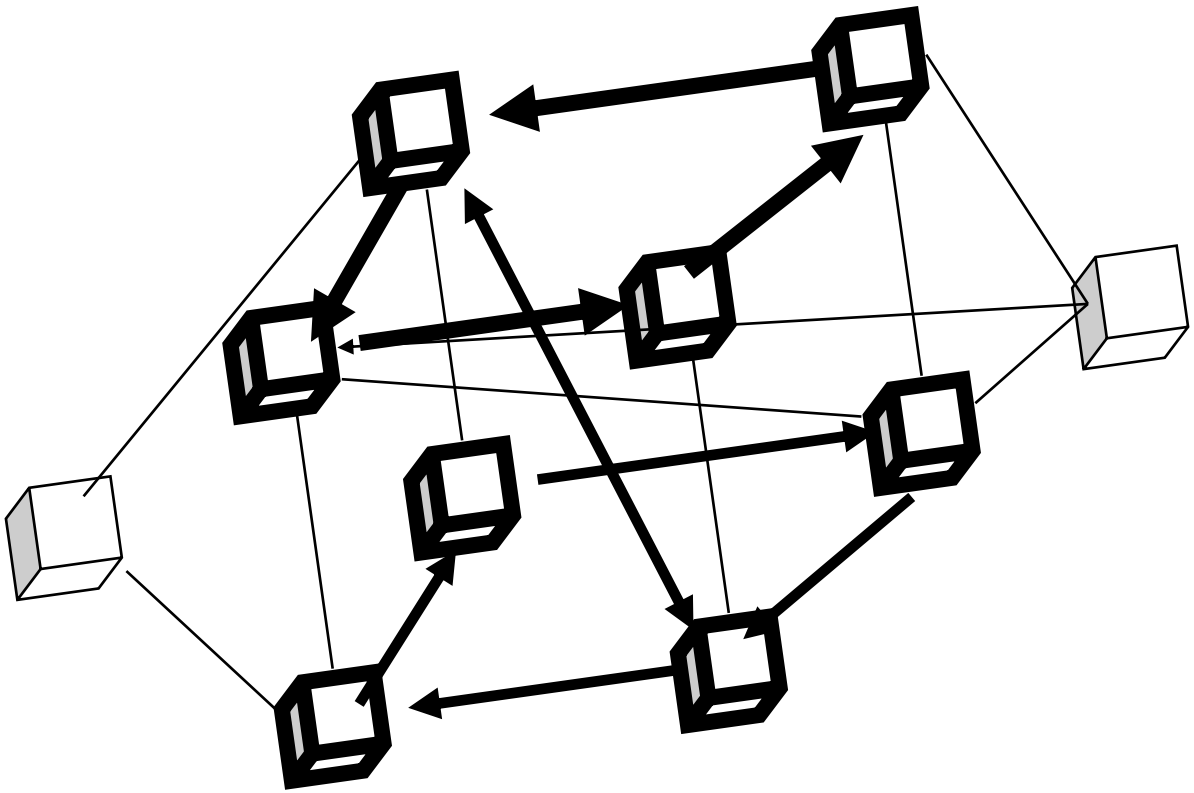
Is Technical Debt a System complexity factor?

Step 4 Identification, Review & Selection of NDI Products

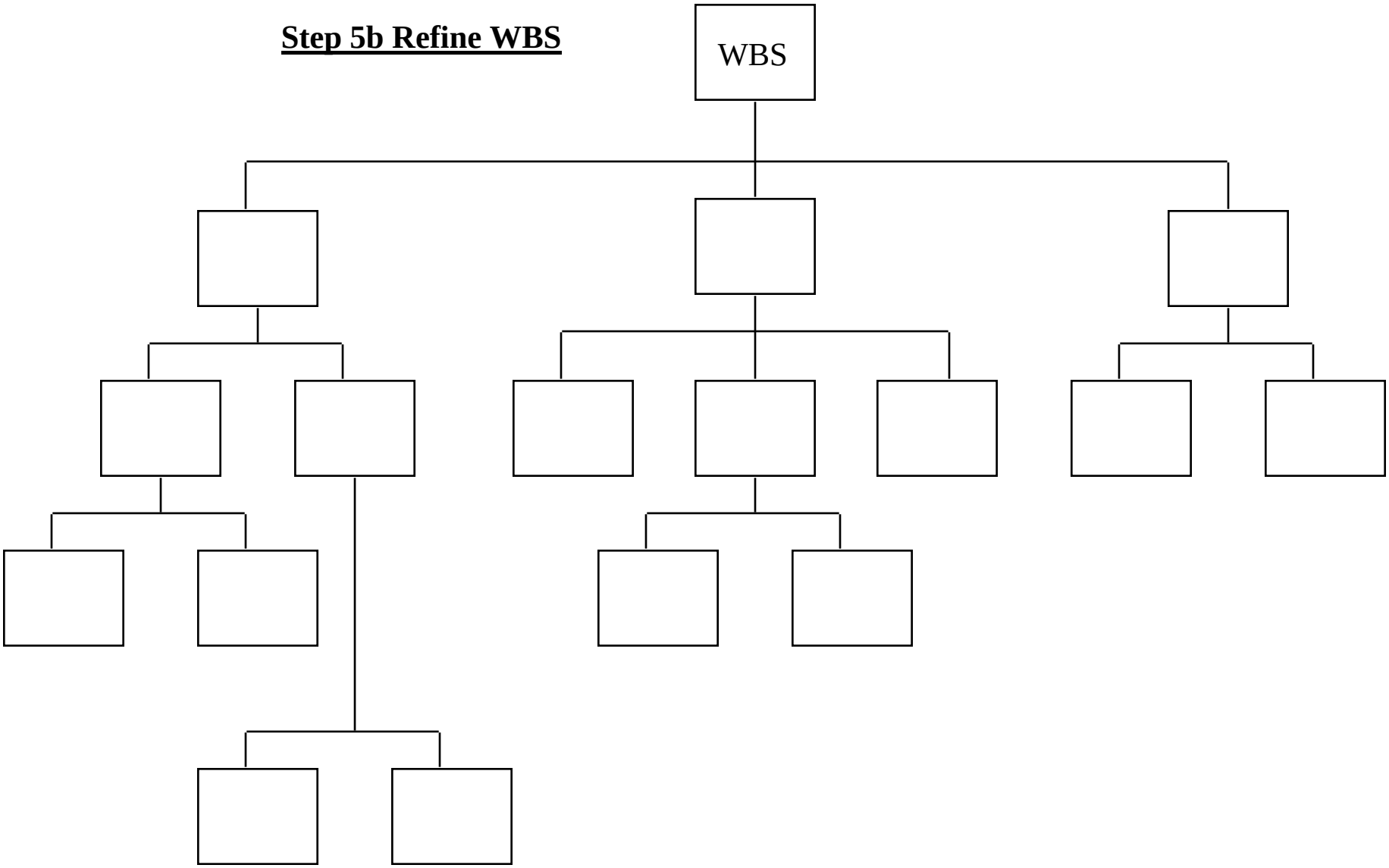


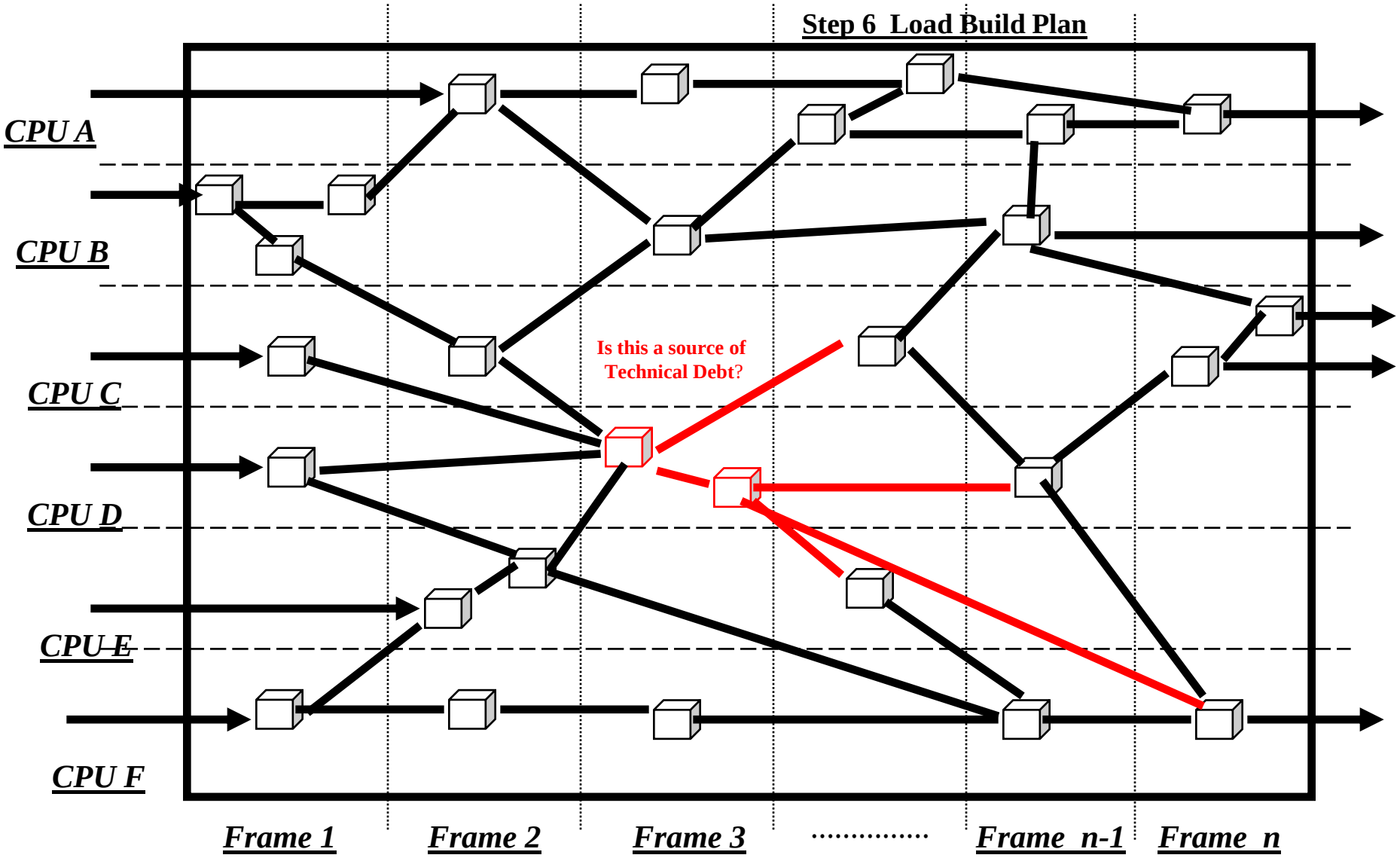
Does selection of NDI incur Technical Debt?

Steps 5a, 8 & 15 Refine Architecture

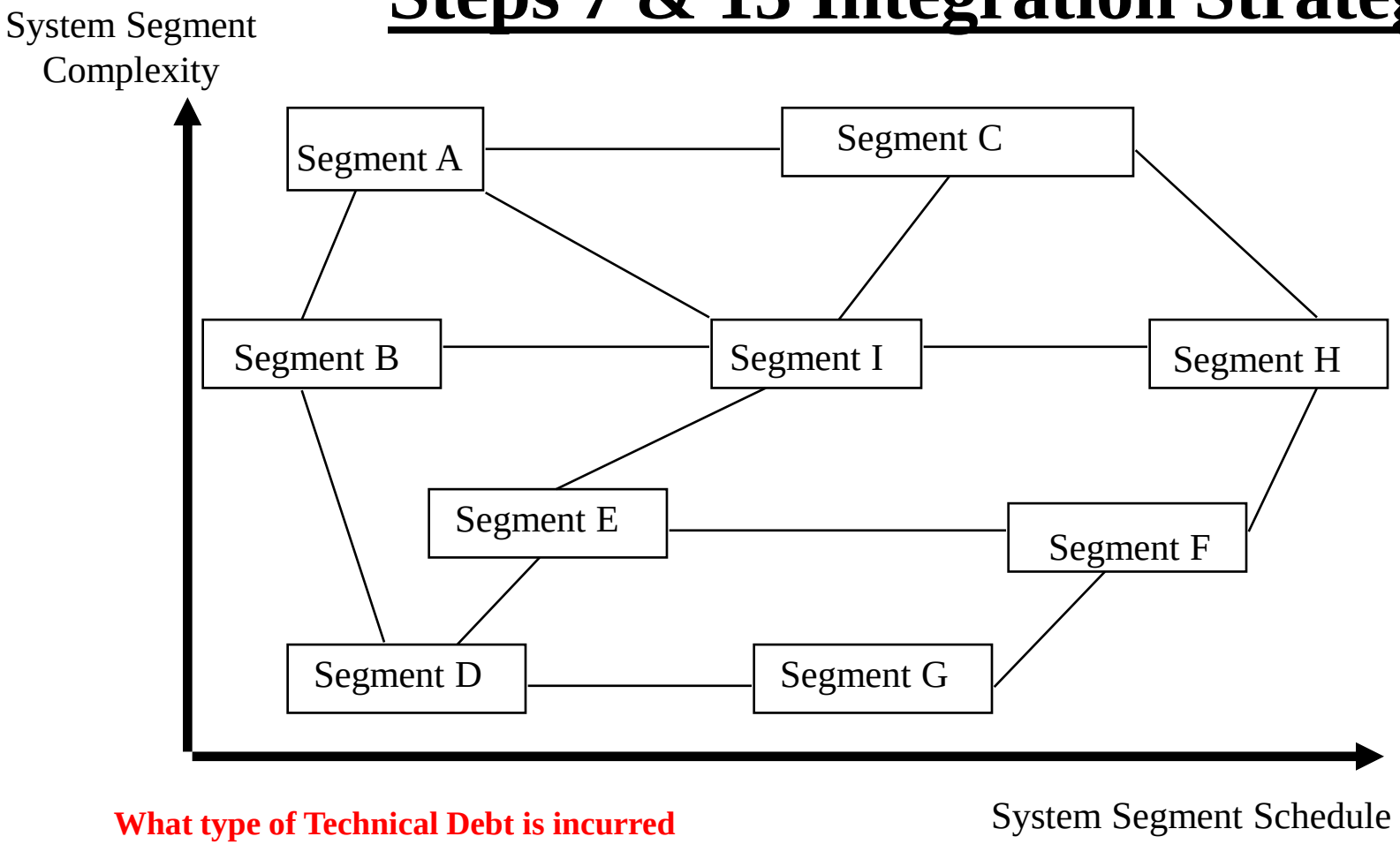


Step 5b Refine WBS




Step 6 Load Build Plan
CPU A
CPU B
CPU C
CPU D
CPU E
CPU F
Is this a source of Technical Debt?
Frame 1
Frame 2
Frame 3
.....
Frame n-1
Frame n

Steps 7 & 13 Integration Strategy/Plan



What type of Technical Debt is incurred if the Integration Strategy is ill-defined?

Step 9 Segment Level Cost Estimate

Cost Estimate				
	Low	Prob	High	Estimate
Analysis				
Design				
⋮				
Integration				
Total				

$$\text{Estimate} = \frac{\text{High} - \text{Low} + 4(\text{Prob})}{6}$$

Step 10 System Level Cost Estimate

Cost Estimate				
	Low	Prob	High	Estimate
Segment A				
Segment B				
⋮				
Segment E				
Integration				

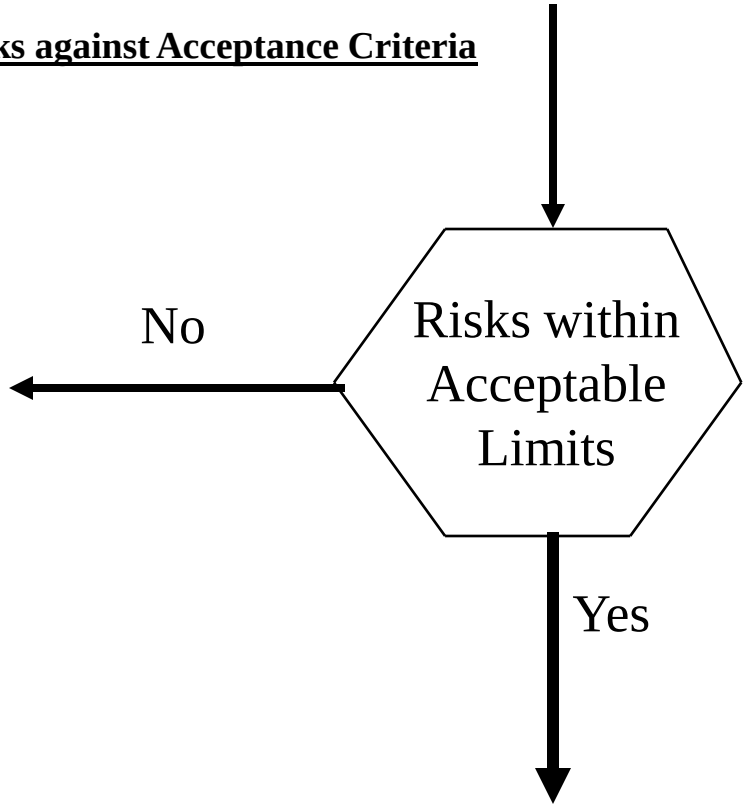
$$\text{Estimate} = \frac{\text{High} - \text{Low} + 4(\text{Prob})}{6}$$

Step 11 Risk Identification/ Mitigation

Risks			
	Sch	Cost	Tech
Risk A	L		
Risk B			H
Risk C	H		
Risk D			L
Risk E		H	

How is Technical Debt related to Schedule, Cost and Technical Risks ?

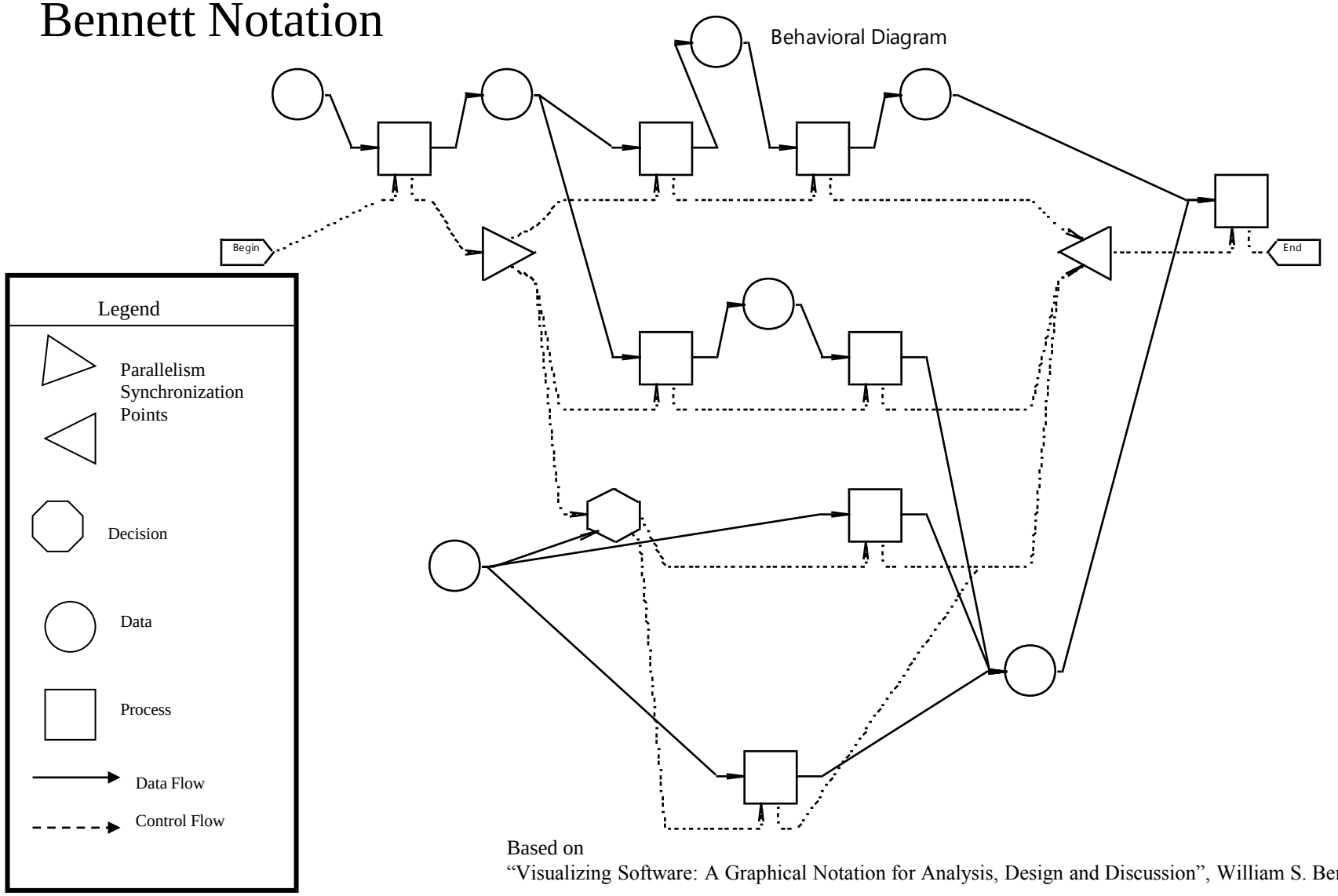
Step 12: Assessment of Risks against Acceptance Criteria



Is this where "intentional" or "unintentional" level of Technical Debt is determined?

Workshop Exercise # 3
System Design & Technical Debt

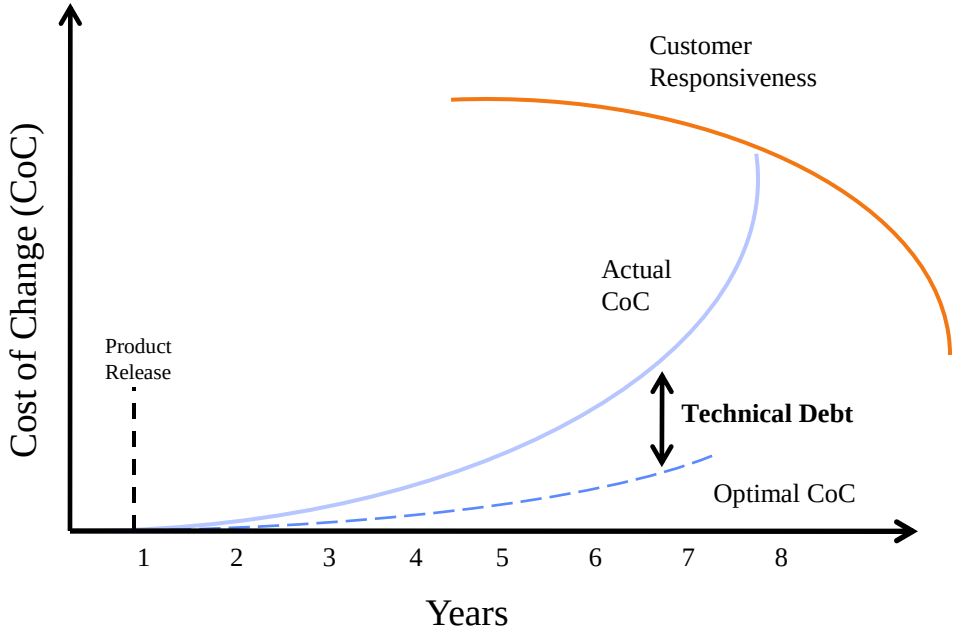
Bennett Notation

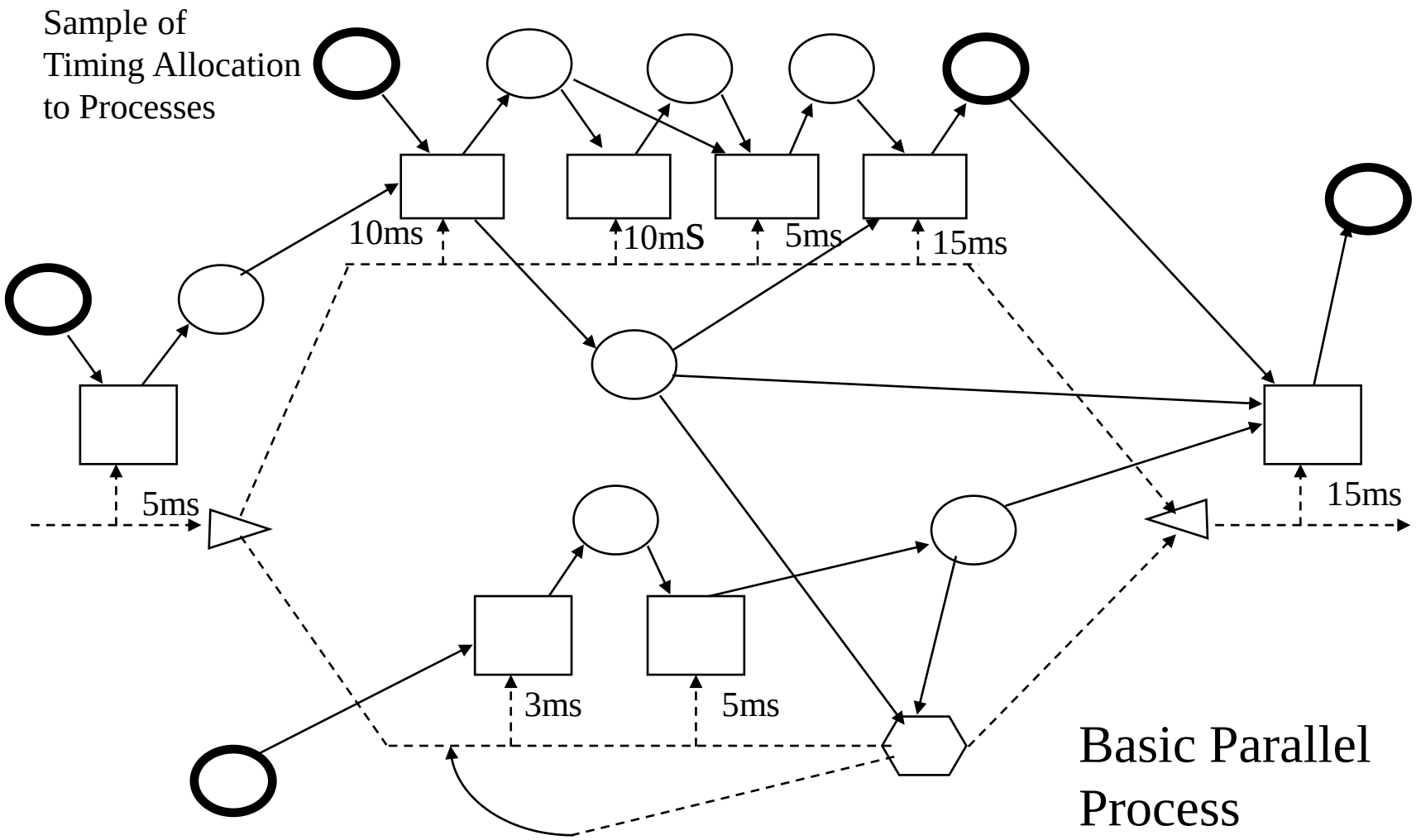


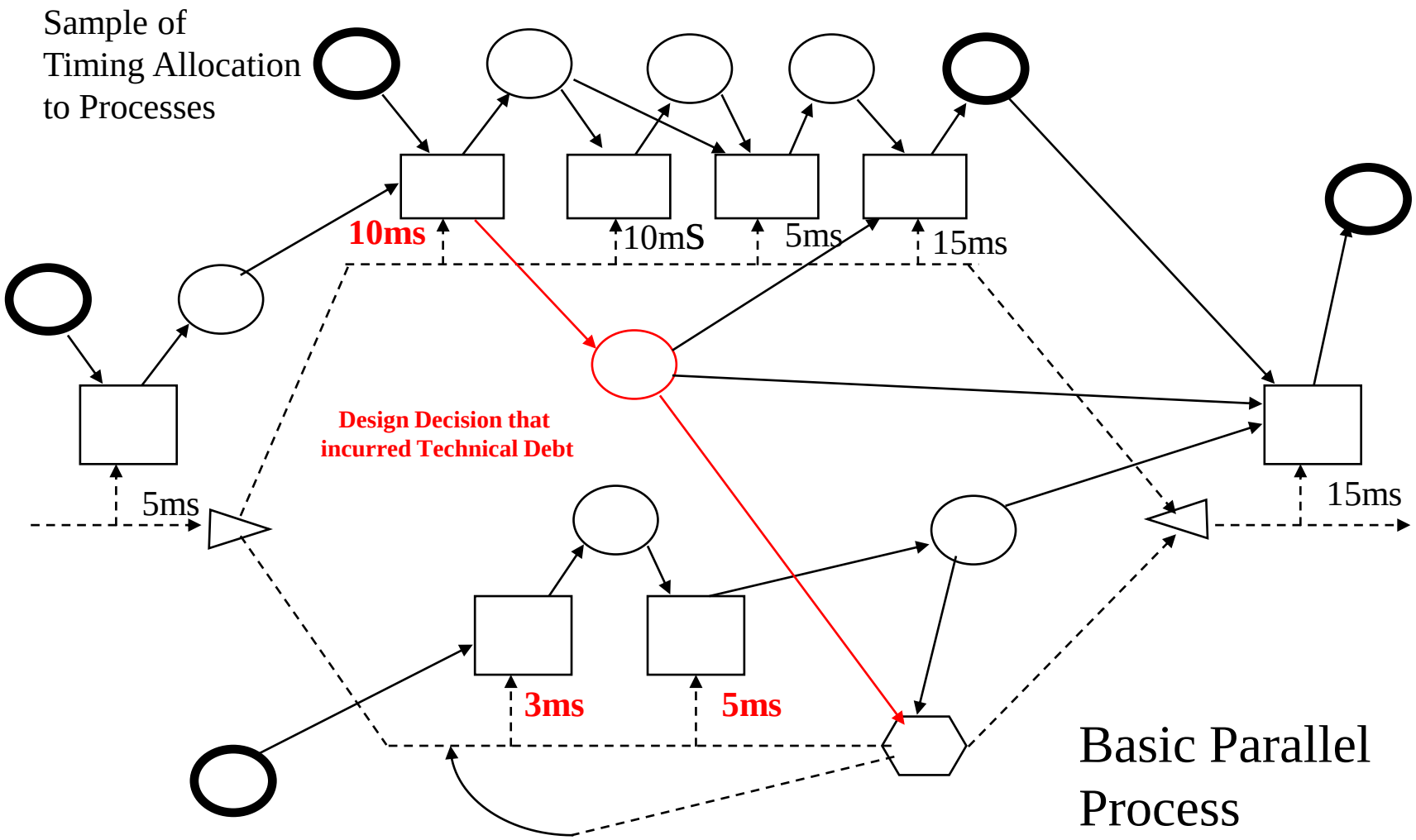
Based on
"Visualizing Software: A Graphical Notation for Analysis, Design and Discussion", William S. Bennett

Technical Debt Observations

"Agile Project Management", Jim Highsmith, second edition



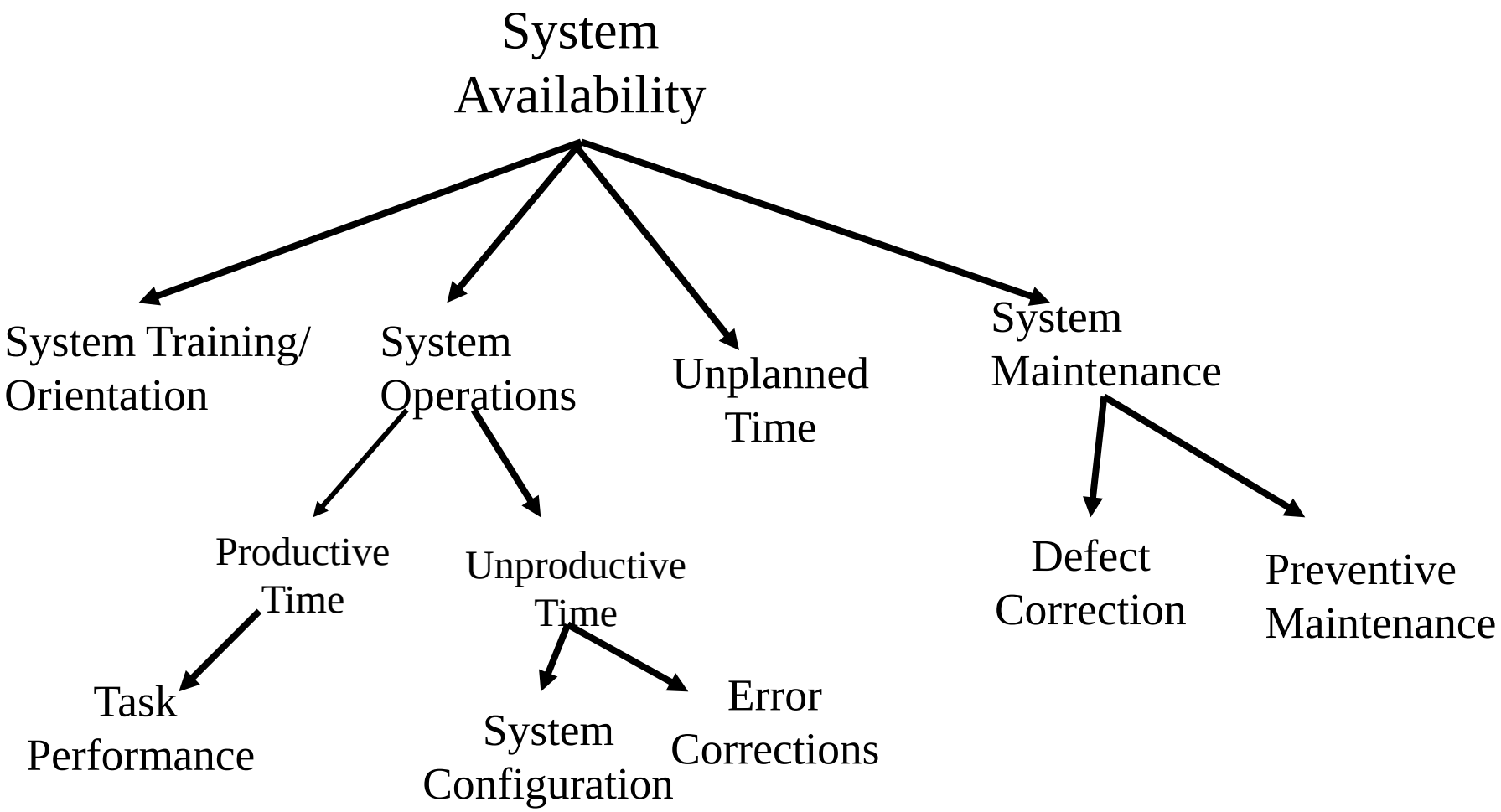

Sample of
Timing Allocation
to Processes
10ms
10mS
5ms
15ms
5ms
15ms
3ms
5ms
Basic Parallel
Process


Sample of
Timing Allocation
to Processes
10ms
10mS
5ms
15ms
Design Decision that
incurred Technical Debt
5ms
15ms
3ms
5ms
Basic Parallel
Process

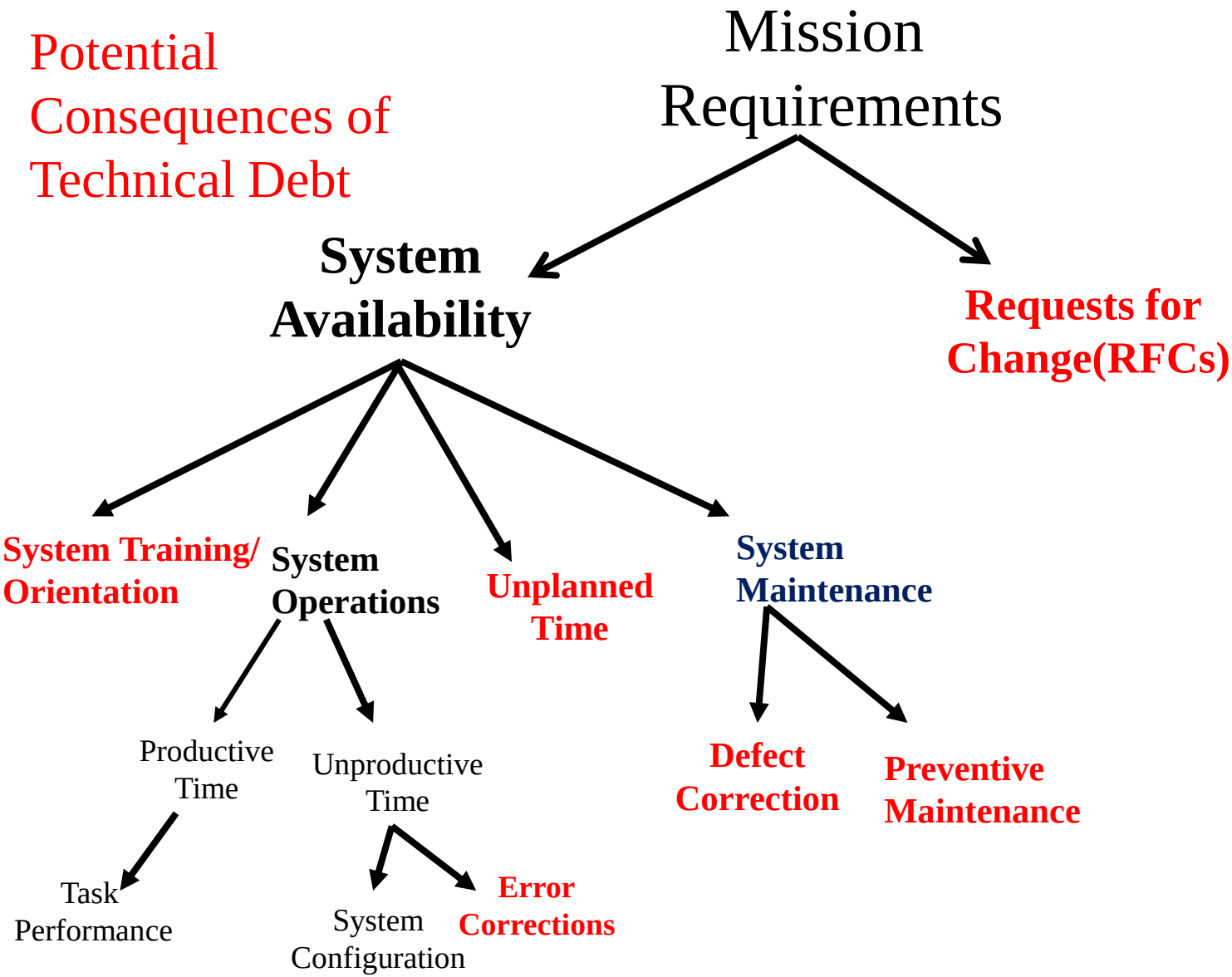
Operations & Maintenance(O&M) Technical Debt

O&M phase occurs after the formal release of the system to the Customer. For DoD contracts, the O&M phase is awarded as a separate Contract. Potentially resulting in a different team executing this phase, then the team that executed the System Development phase. O&M phase is cited as the phase where significant costs occur and therefore is a target for Affordability activities to address.

Commercial products have a similar phase which may determine how much of the market share that a company captures with the release of new versions of their system, based on competitive pressures and consumer demands.

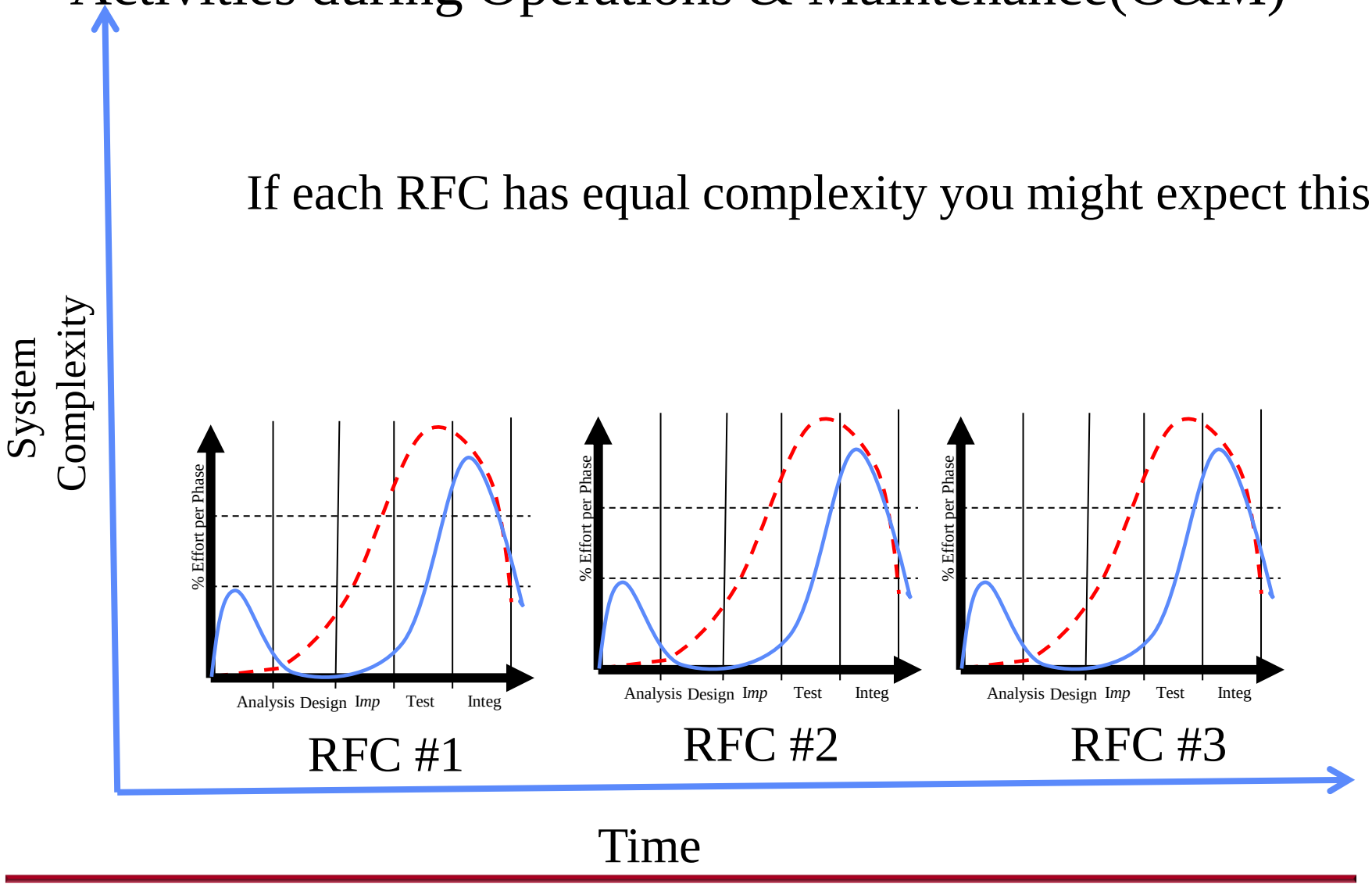

System Availability
System Training/ Orientation
System Operations
Unplanned Time
System Maintenance
Productive Time
Unproductive Time
Defect Correction
Preventive Maintenance
Task Performance
System Configuration
Error Corrections

Potential Consequences of Technical Debt



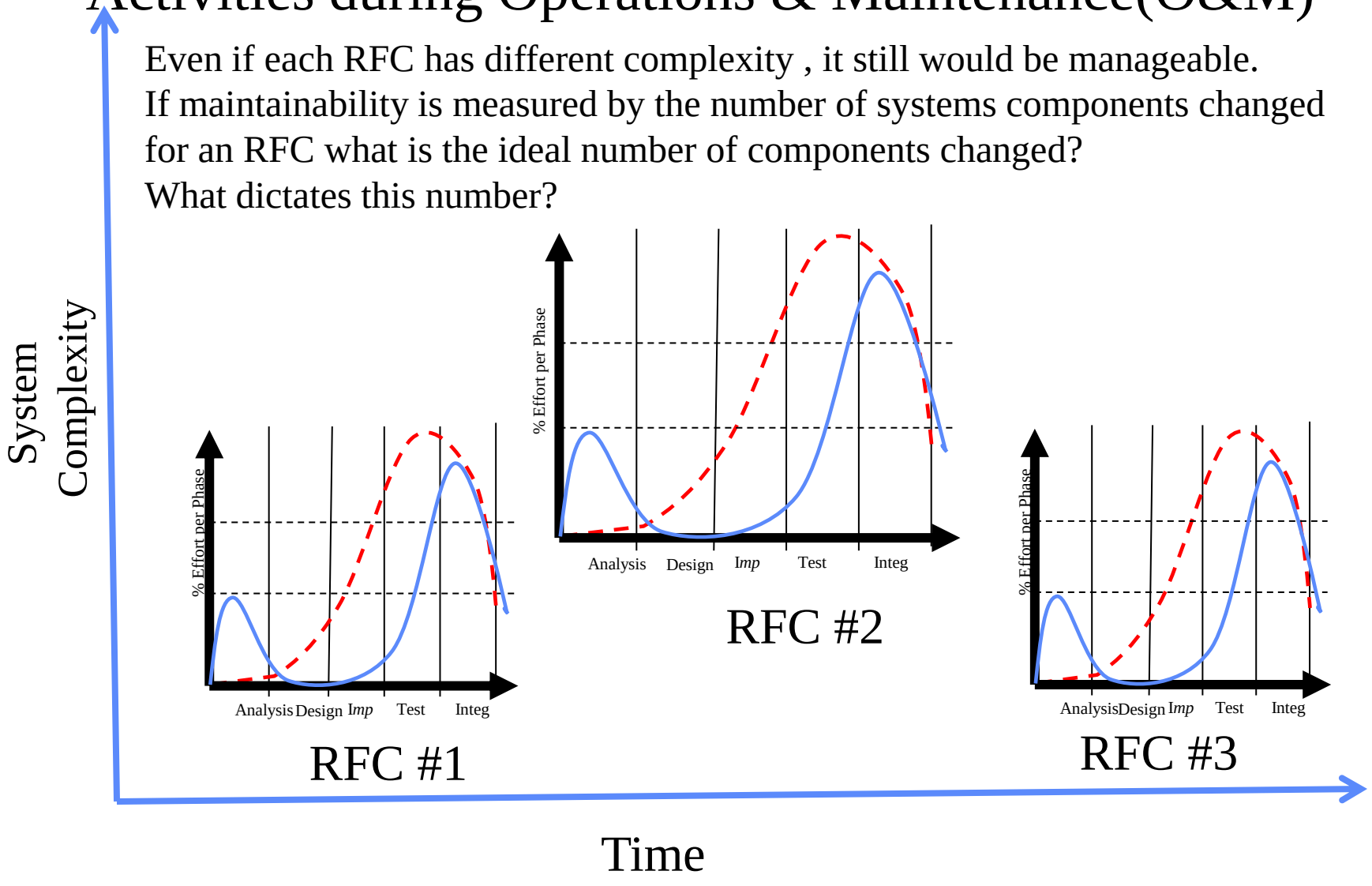
Activities during Operations & Maintenance(O&M)

If each RFC has equal complexity you might expect this



Activities during Operations & Maintenance(O&M)

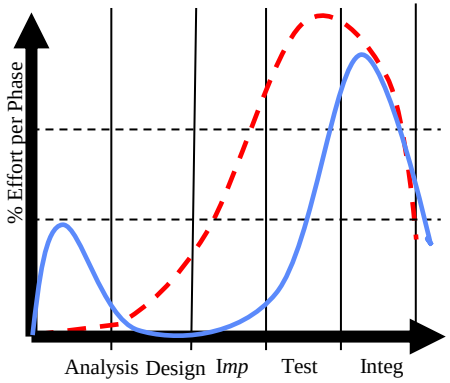
Even if each RFC has different complexity , it still would be manageable.
If maintainability is measured by the number of systems components changed for an RFC what is the ideal number of components changed?
What dictates this number?



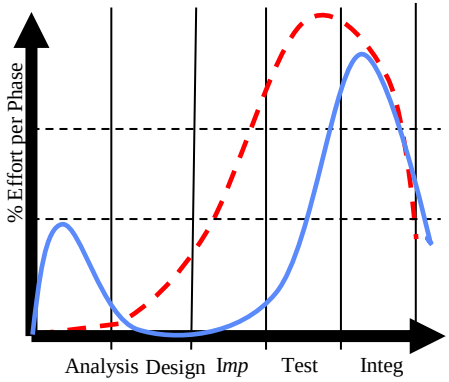
Activities during Operations & Maintenance(O&M)

However, experience shows that even when the RFCs have the same complexity the "aggregate" complexity of the system causes side effects which complicate future RFCs.

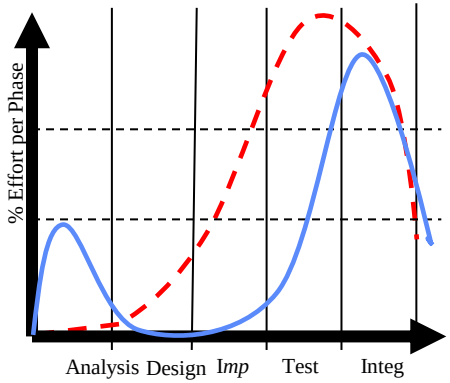
System Complexity



RFC #1



RFC #2

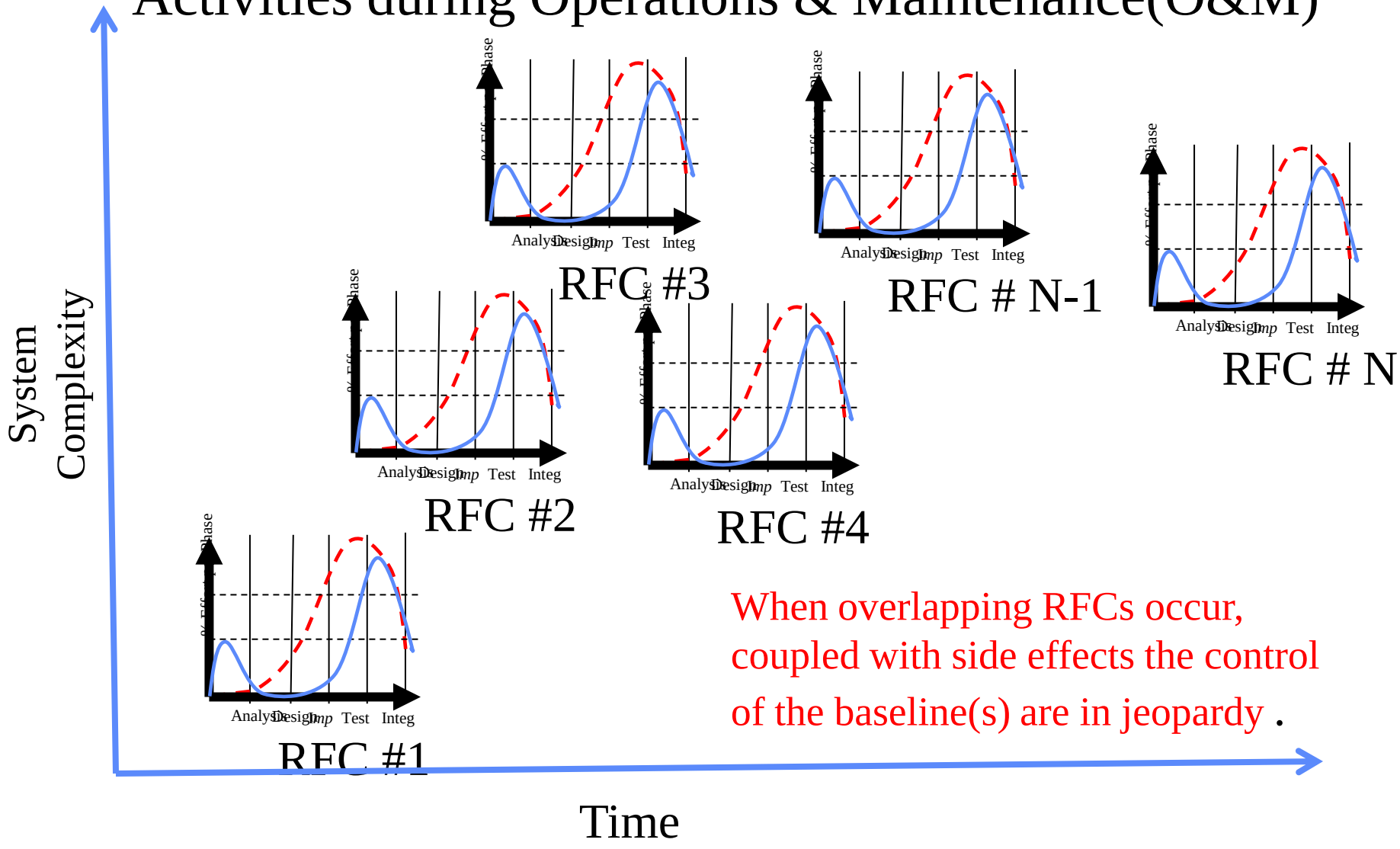


RFC #3

How is the phenomenon managed?

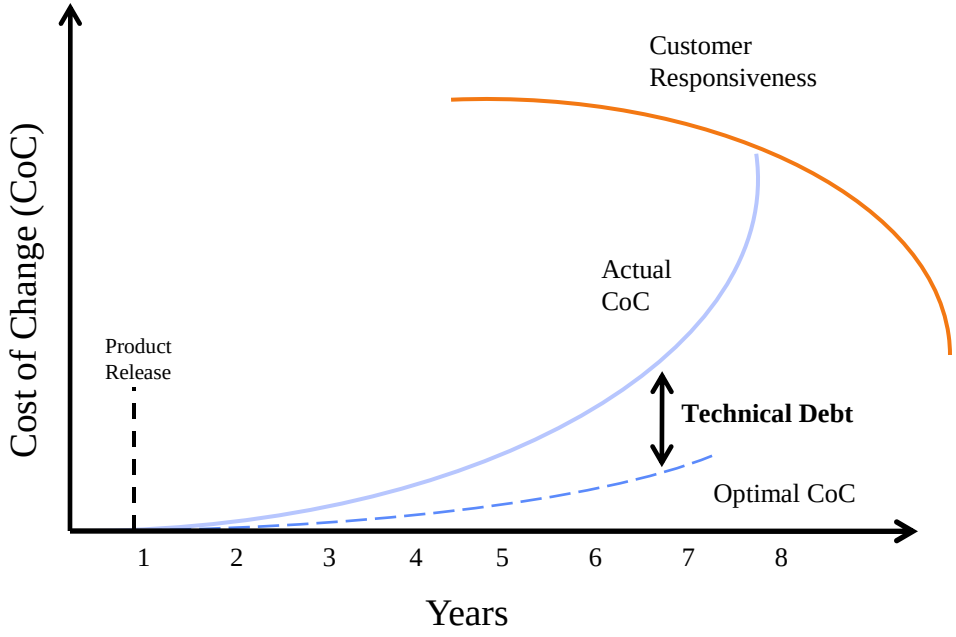
Time

Activities during Operations & Maintenance(O&M)

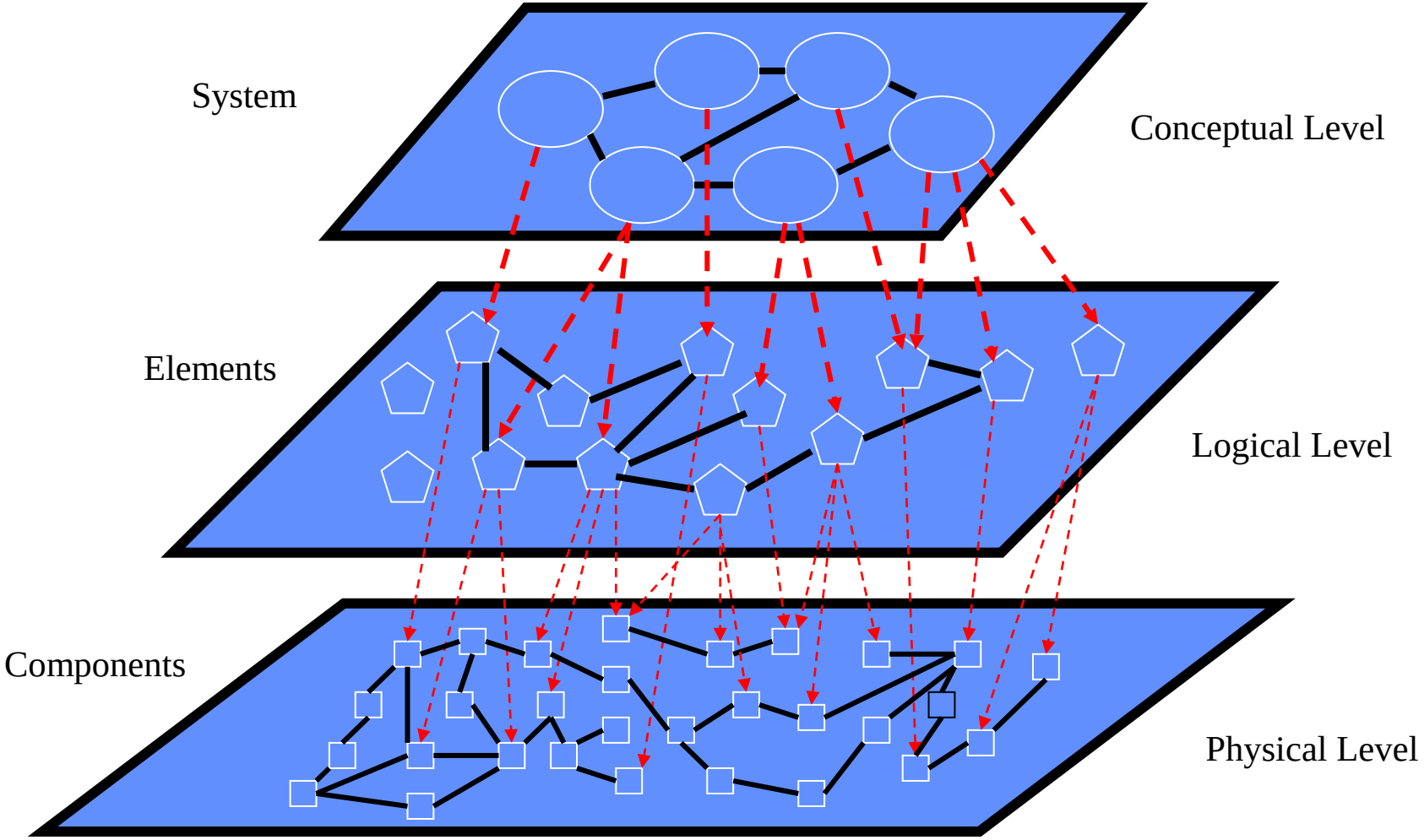


Technical Debt Observations

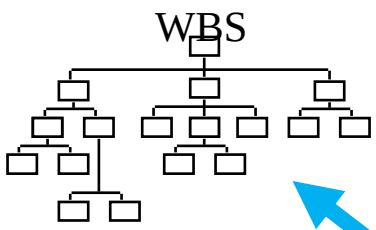
"Agile Project Management", Jim Highsmith, second edition

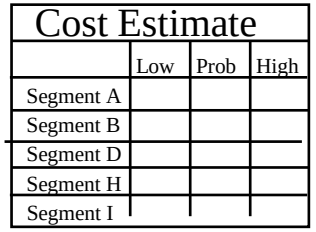


Levels of Architecture
(Conceptual to Logical to Physical Mapping)

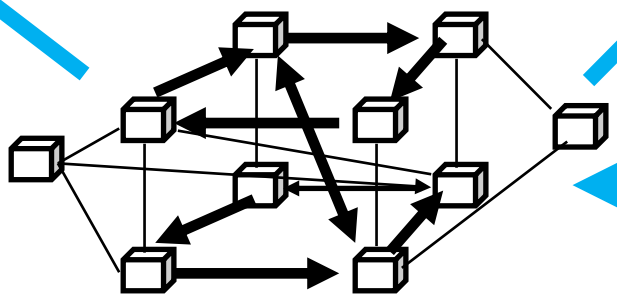


Notional Architecture Process

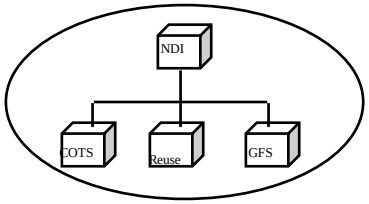




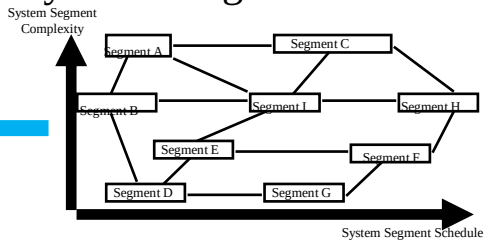
Architecture



NDI Products

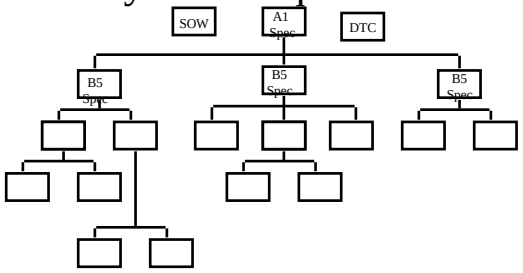


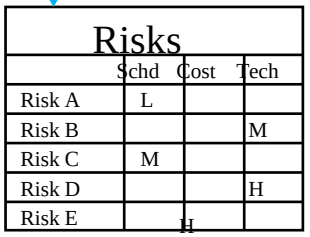
System Integration Plan



Business Requirements

System Requirements

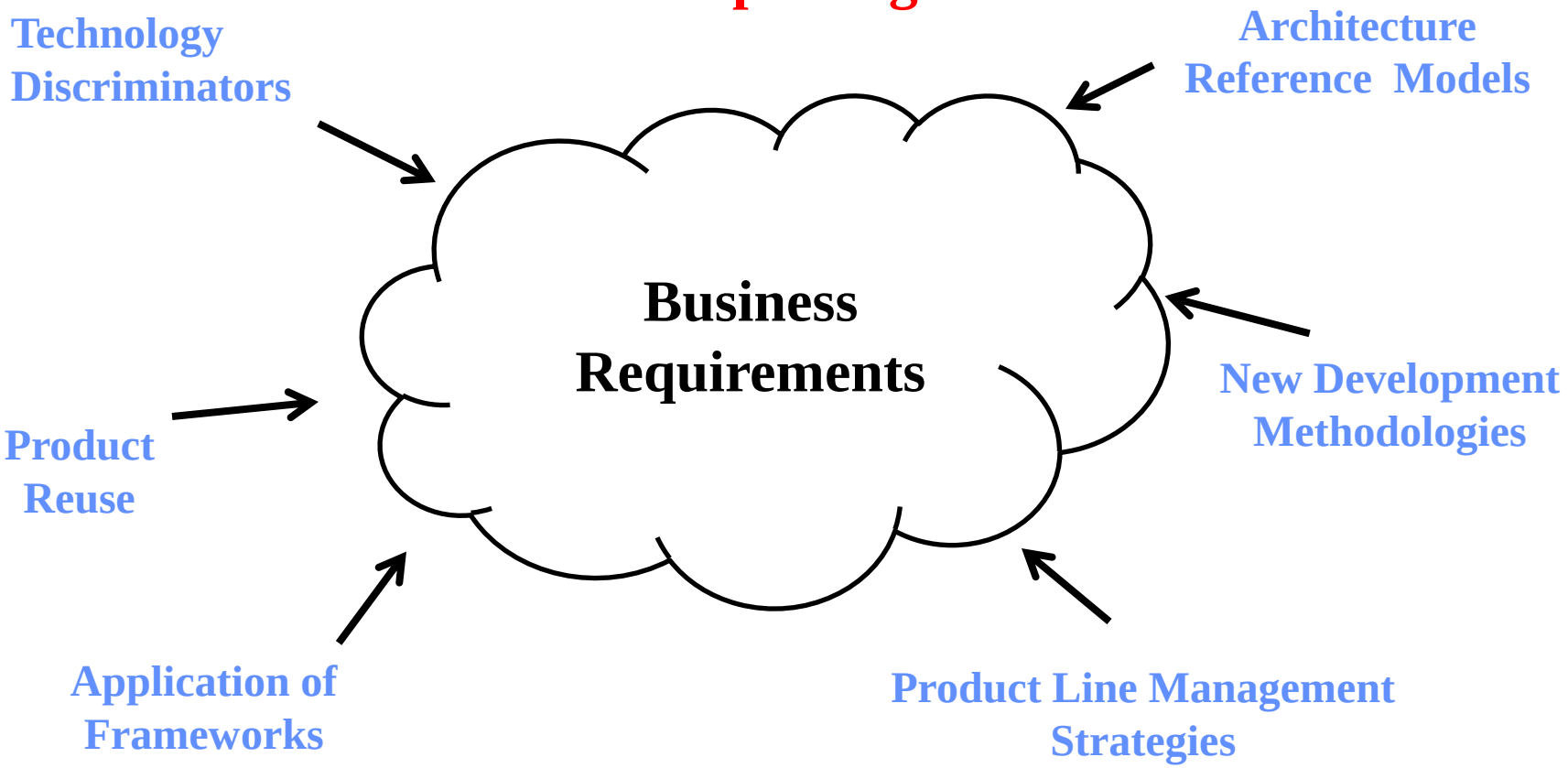




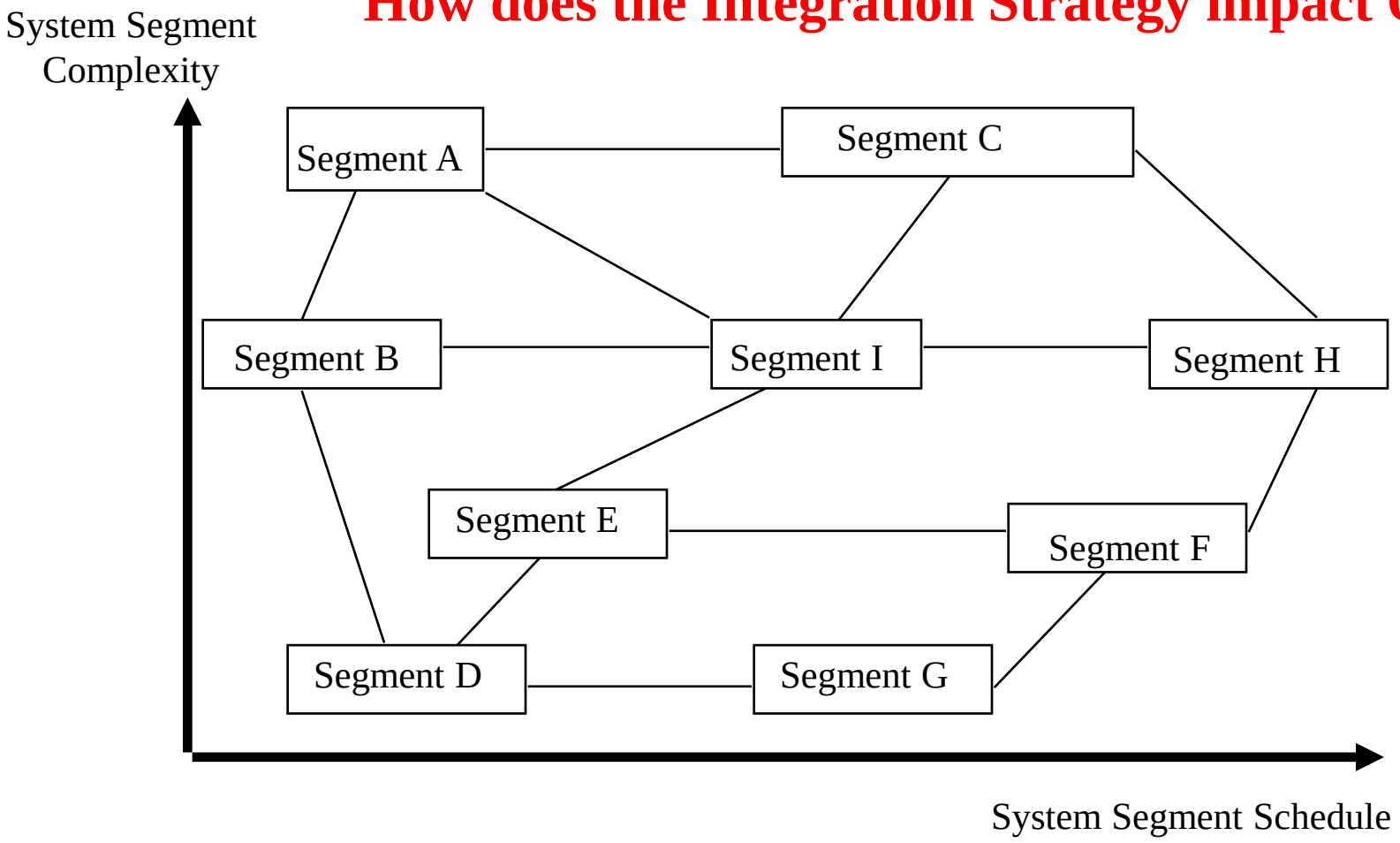
(Domain Knowledge)

Where does Technical Debt be incurred in this process causing impacts in O&M?

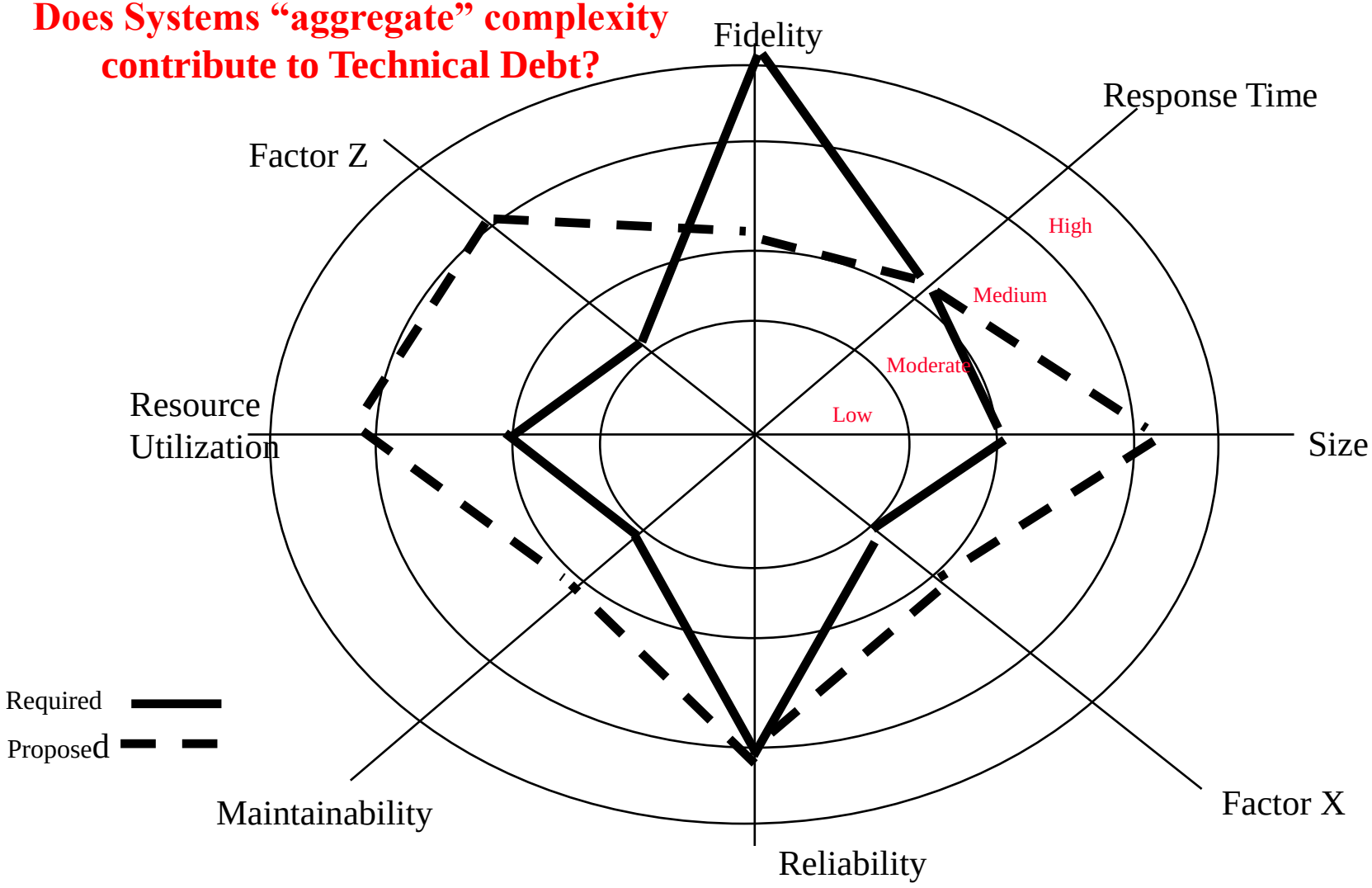
When does this become a Source of Technical Debt impacting O&M?

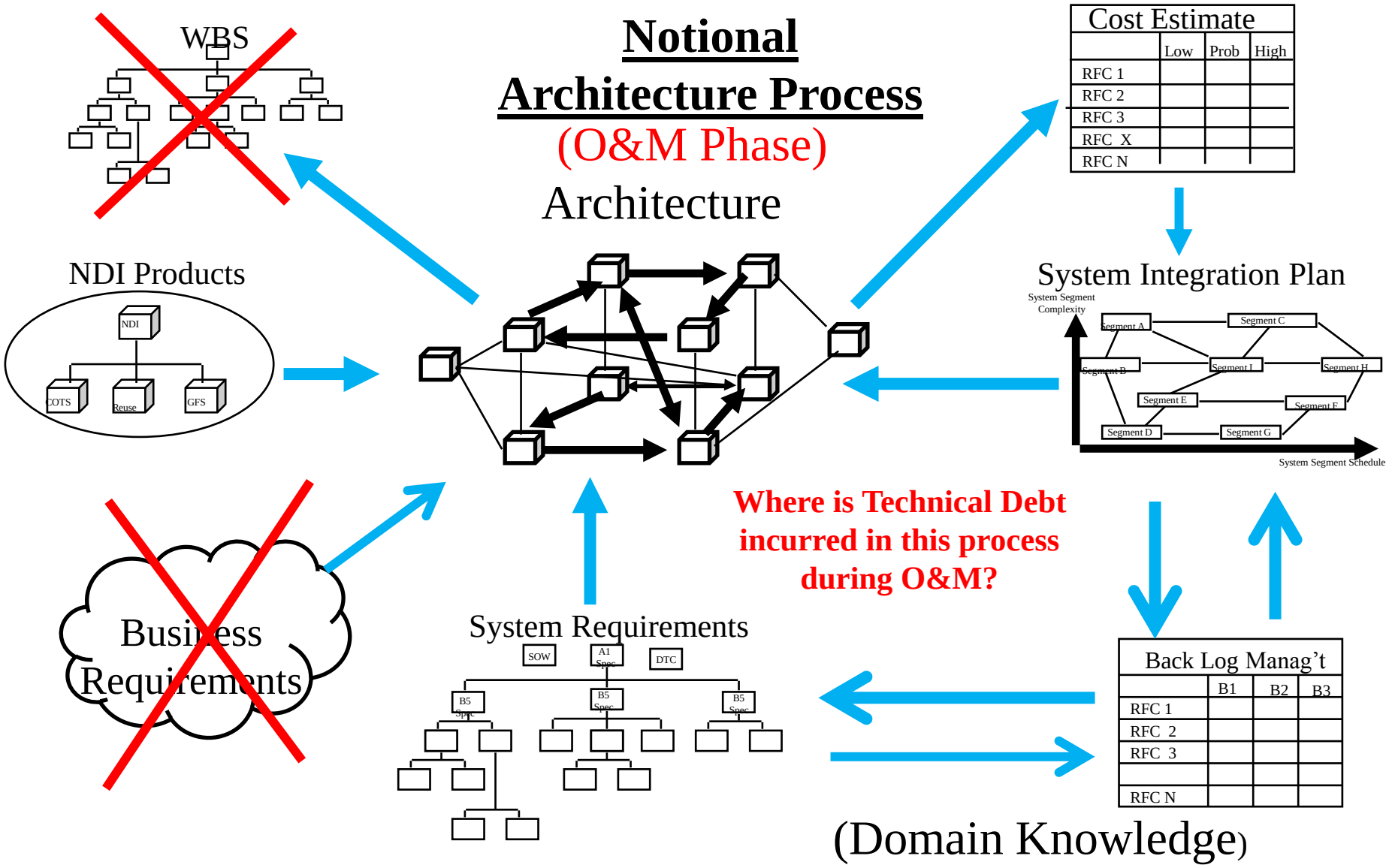


How does the Integration Strategy impact O&M?

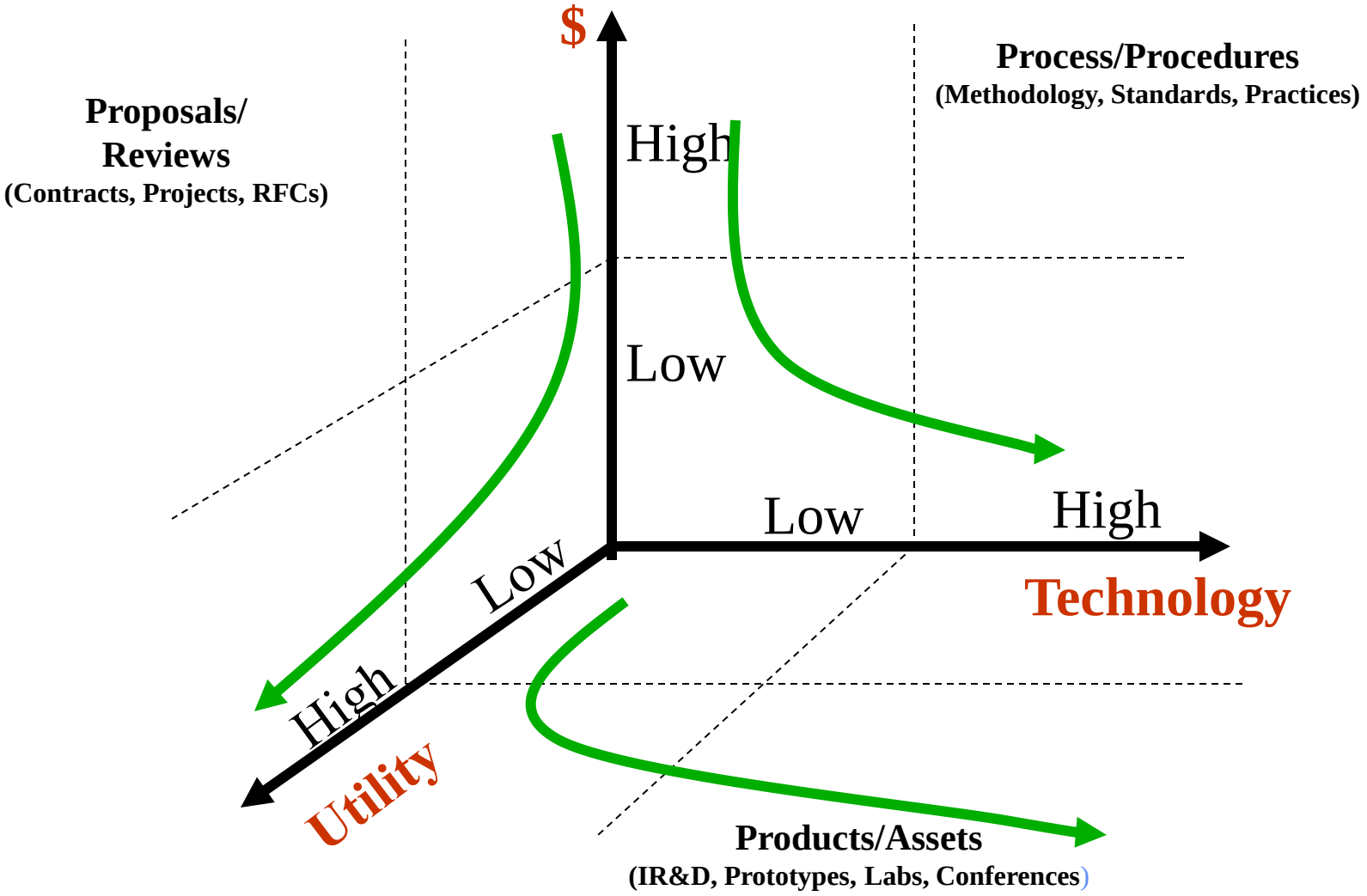


Does Systems "aggregate" complexity contribute to Technical Debt?




Notional
Architecture Process
(O&M Phase)
Architecture
WBS
Cost Estimate
Low
Prob
High
RFC 1
RFC 2
RFC 3
RFC X
RFC N
NDI Products
NDI
COTS
Reuse
GFS
System Integration Plan
System Segment Complexity
Segment A
Segment C
Segment B
Segment I
Segment H
Segment E
Segment F
Segment D
Segment G
System Segment Schedule
Where is Technical Debt incurred in this process during O&M?
Business Requirements
System Requirements
SOW
A1 Spec
DTC
B5 Spec
Back Log Manag't
B1
B2
B3
RFC 1
RFC 2
RFC 3
RFC N
(Domain Knowledge)

"The Challenge"



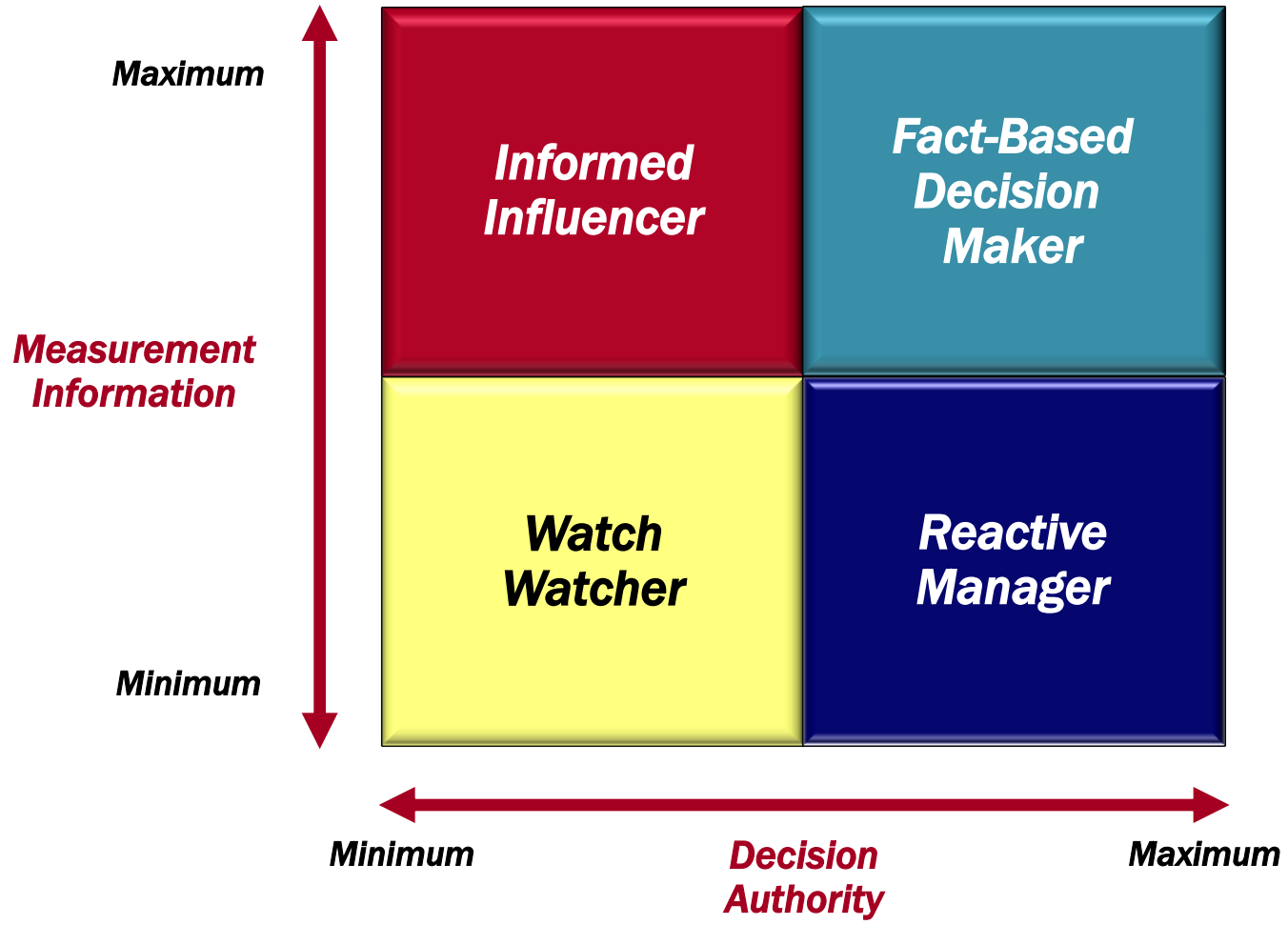
Measurement/Management of Technical Debt

The key to any concern during System Development and O&M is how the concern is measured and managed. Information provided by PSM provides the tools to perform both function.

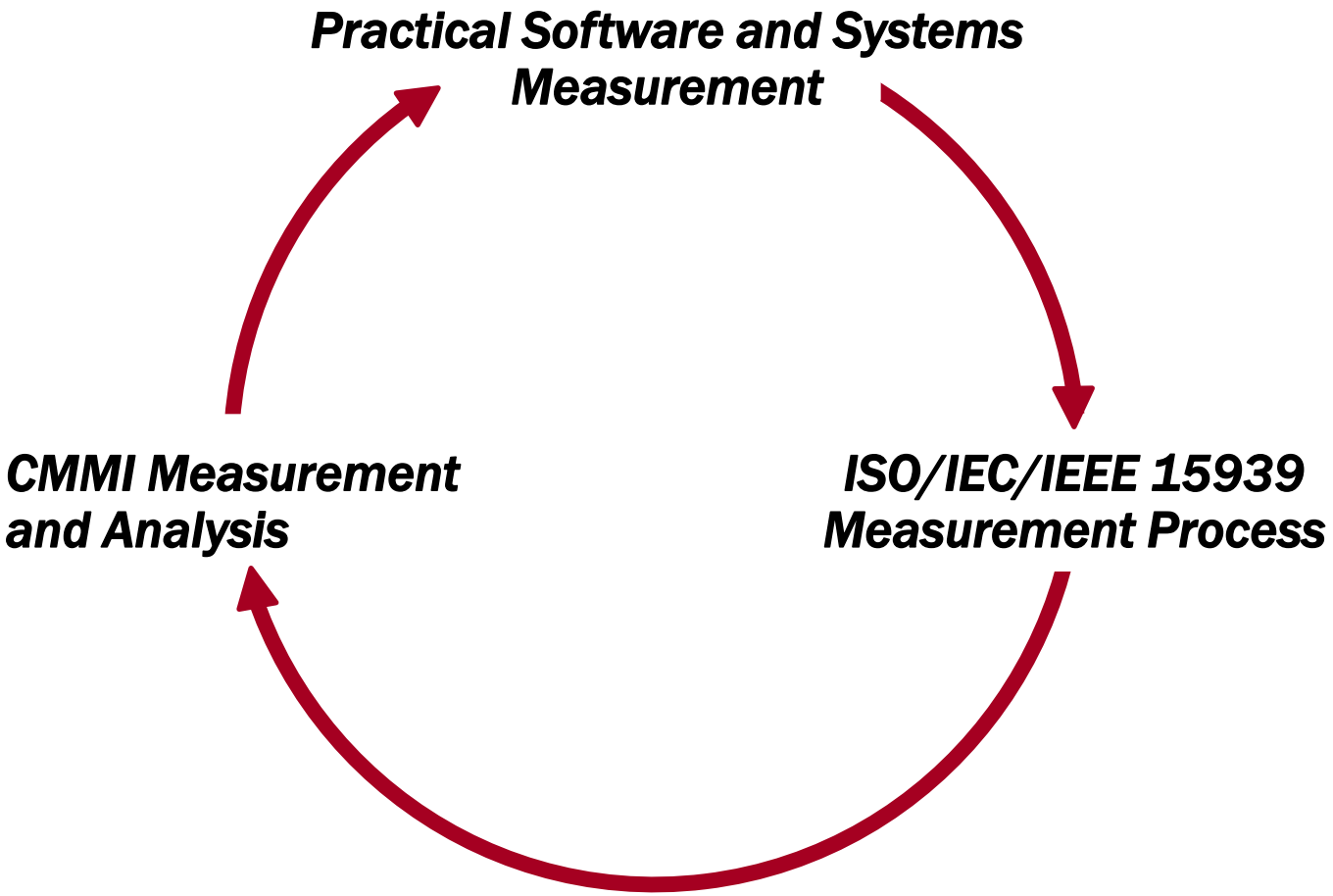
"Intentional" Technical Debt must be captured at its source in order to access it throughout the development and O&M phases of the System Lifecycle.

The following provides the framework for performing the measurement & management of Technical Debt. This is of particular concern when the organization executing the O&M phase is not the organization that executed the System Development phase.

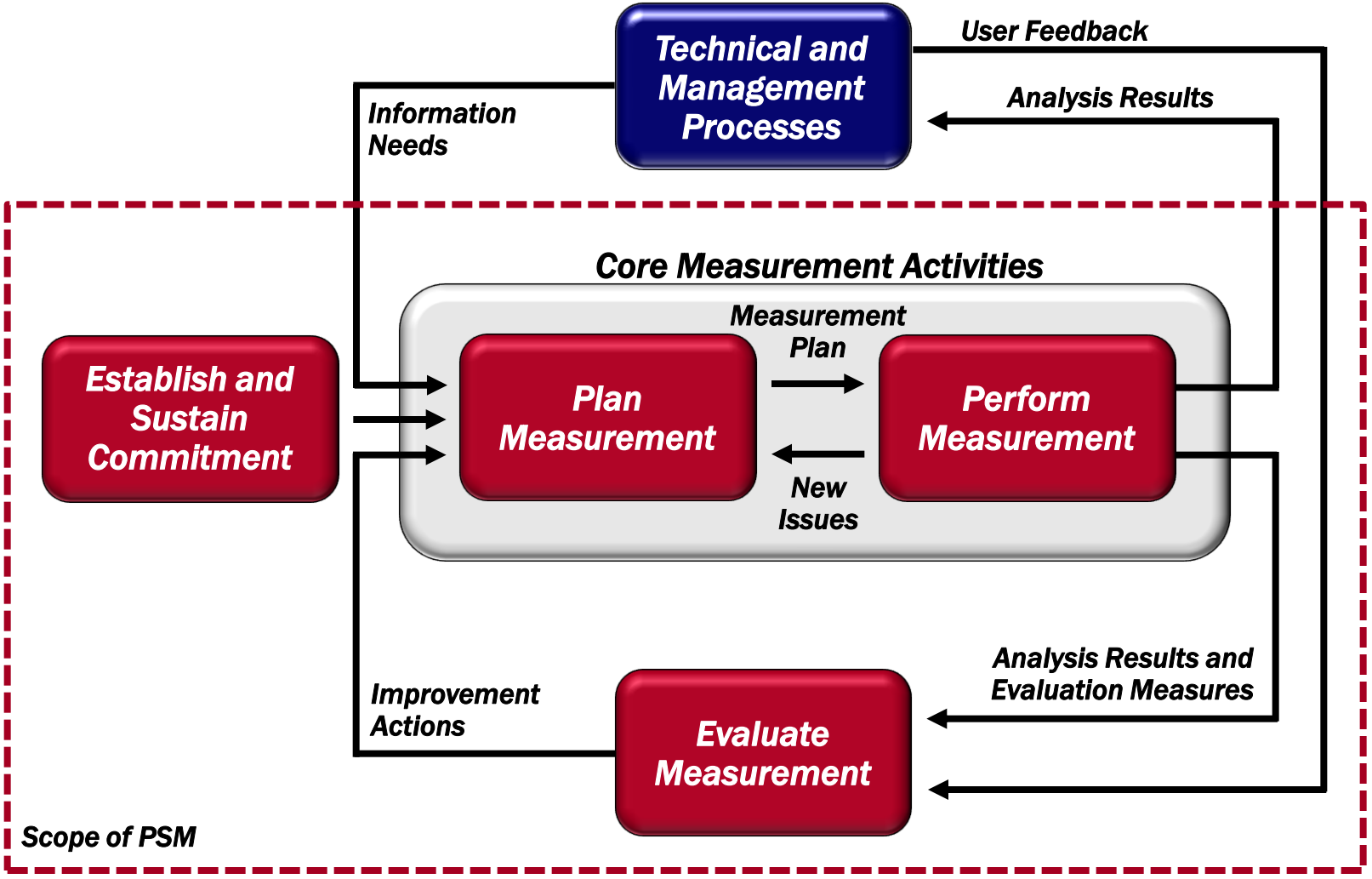
Measurement and Decision Making



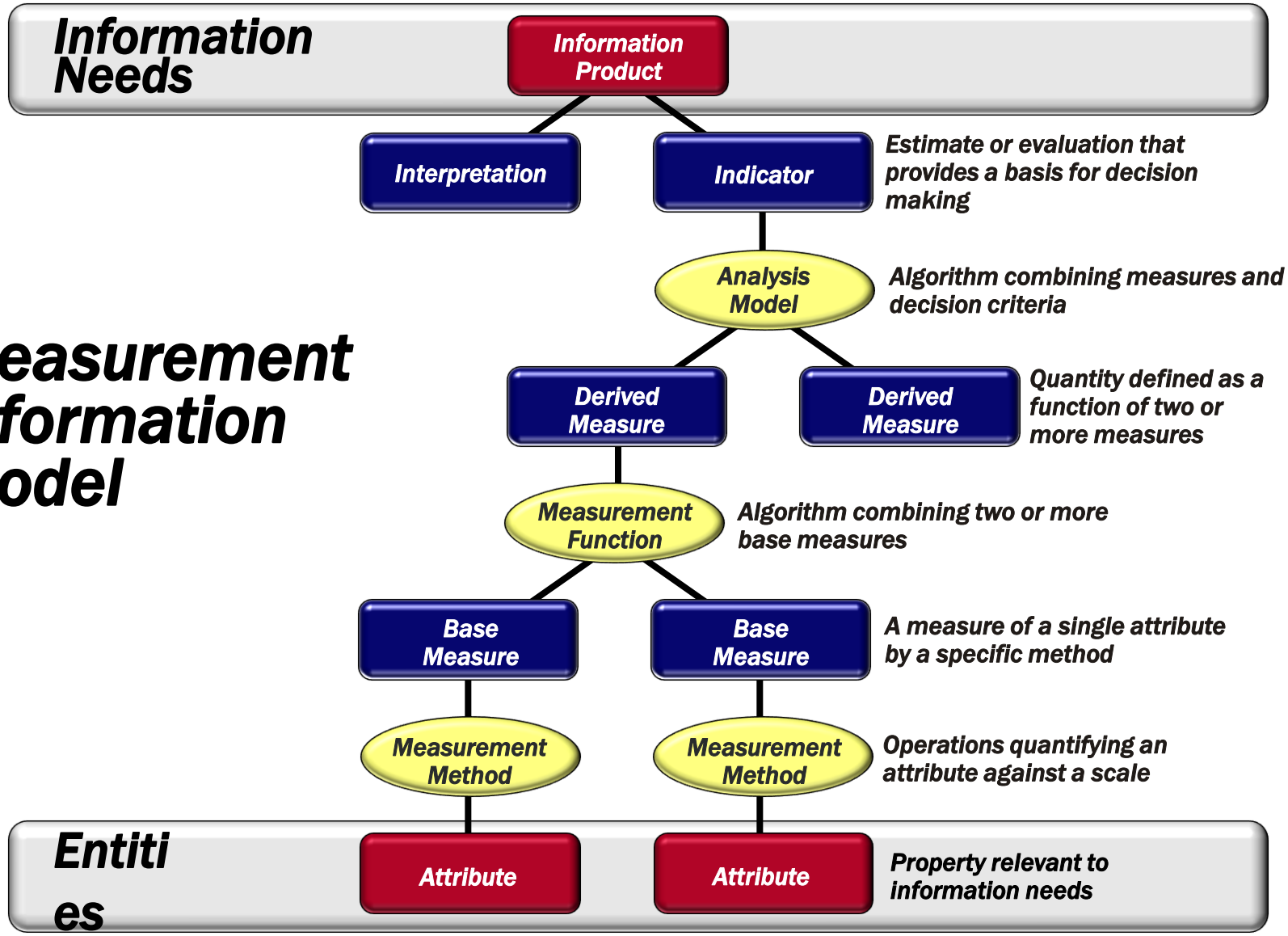
PSM - International Standards - CMMI



Measurement Process Model



Measurement Information Model



Performance Changing Decisions?

- *Add more people*
- *Build components in parallel*
- *Ignore development dependencies*
- *Schedule "backwards" from the delivery date*
- *Incrementally defer functionality*
- *Relax process requirements*
- *Postpone rework*
- *Minimize functional testing*
- *Redefine exit criteria*
- *Reduce requirements*

System Performance Issues

Identified Issues	Relative Occurrence
Process Capability	91 %
Organizational Management	87 %
Requirements Management	87 %
Product Testing	83 %
Program Planning	74 %
Product Quality - Rework	70 %
System Engineering	61 %
Process Adherence	52 %
Program Schedule	48 %
Interoperability	43 %
...	
Configuration Management	26%

Reference: Results from TAI systemic analysis of 50 large-scale DoD projects.

Source: "Introduction to PSM Course, Version 7.0", July 30,2012, Portsmouth, VA

Information Categories - Measurable Concepts

Schedule and Progress
- Milestone Completion
- Work Unit Progress
- Work Backlog
- Incremental Capability

Resources and Cost
- Financial Performance
- Personnel Effort
- Environment and Support Resources

Size and Stability
- Physical Size and Stability
- Functional Size and Stability

Product Quality
- Functional Correctness
- Supportability - Maintainability
- Efficiency
- Portability
- Usability
- Dependability - Reliability
- Security - Safety

Process Performance
- Process Compliance
- Process Efficiency
- Process Effectiveness

Technology Effectiveness
- Technology Suitability
- Technology Maturity
- Technology Volatility
- Technology Adoption

Customer Satisfaction
- Customer Feedback
- Customer Support

Source: "Introduction to PSM Course, Version 7.0", July 30,2012, Portsmouth, VA

Information Categories - Measurable Concepts

Schedule and Progress
- Milestone Completion
- Work Unit Progress
- Work Backlog
- Incremental Capability

Resources and Cost
- Financial Performance
- Personnel Effort
- Environment and Support Resources

Size and Stability
- Physical Size and Stability
- Functional Size and Stability

Product Quality
- Functional Correctness
- Supportability - Maintainability
- Efficiency
- Portability
- Usability
- Dependability - Reliability
- Security - Safety

Measurable Concept Areas where "Intentional" Technical Debt may be found

Process Performance
- Process Compliance
- Process Efficiency
- Process Effectiveness

Technology Effectiveness
- Technology Suitability
- Technology Maturity
- Technology Volatility
- Technology Adoption

Customer Satisfaction
- Customer Feedback
- Customer Support

Source: "Introduction to PSM Course, Version 7.0", July 30,2012, Portsmouth, VA

Technical Debt Workshop Summary

- *Technical Debt*
  - *Management of Technical Debt- Steve McConnell*
  - *Technical Debt Observations- Jim Highsmith*
  - *Types of Debt- Chris Sterling*
  - *Management of Architectural Debt- Ipek Ozkapa*
- *Technical Debt applicability to Systems*
  - *Workshop Exercise # 1-Identifying Technical Debt*
  - *Workshop Exercise # 2- Architecture & Technical Debt*
  - *Workshop Exercise # 3- System Design & Technical Debt*
  - *Impact of Technical Debt on Operations & Maintenance (O&M)*
- *Measurement/Management of Technical Debt*

Technical Debt References

- *Managing Technical Debt*
  - *Tom McConnell*
  - *http://www.construx.com/Page.aspx?cid=2801*
- *The Financial Implications of Technical Debt*
  - *Jim Highsmith*
  - *http://www.jimhighsmith.com/2010/10/19/the-financial-implications-of-technical-debt/*
- *Second International Workshop on Managing Technical Debt*
  - *http://www.sei.cmu.edu/community/td2011/*
- *Technical Debt or Strategic Opportunity*
  - *https://files.me.com/philippe.kruchten/rgw078*
- *Enabling Agility by Strategically Managing Architectural Technical Debt*
  - *Ipek Ozkaya*
  - *http://blog.sei.cmu.edu/post.cfm/enabling-agility-by-strategically-managing-architectural-technical-debt*

- *Practical Software Measurement: Objective Information for Decision Makers"*
  - *J. McGarry, D. Card, C. Jones, B. Layman, E. Clark, J. Dean, F. Hall*
  - *Addison-Wesley, Boston, 2002*

- *Introduction to PSM Course, Version 7.0, July 30,2012, Portsmouth, VA*

Technical Debt Workshop Action Plan

- *White paper on Technical Debt applicability to Systems*
  - *Describe Technical Debt in Systems Engineering Vernacular*
  - *Identify sources and methods of measurement of Technical Debt within the Systems Engineering Life Cycle*
    - *Systems Analysis*
    - *Systems Architecture*
    - *Systems Design*
    - *Systems Build*
    - *Systems Integration & Test*
    - *System Operation & Maintenance*
  - *Identify of implication of System Level Technical Debt to Software and Hardware Elements*
  - *Measurement/Management of Technical Debt*

Practical Software and Systems Measurement

Objective Information for Decision Makers

What Does Technical Debt Mean at a System Level (Outbrief)

2 August 2012

Bob Epps, Lockheed Martin

What Does Technical Debt Mean at a System Level

- *Identification and Management of Technical Debt in System Development and O&M phases of the Systems Lifecycle.*

- *Discuss how the decisions made during the System life cycle influence the level of Technical Debt of in the System.*

- *This will include the evaluation of Technical Debt's impact to the Operation & Maintenance (O&M) phase.*

Workshop Participants

- Garry Roedler
- Jim Stubbe
- Mauricio Aguiar
- Jim McCurey
- Jeff Howard
- Paul McMahon
- Cheryl Garrison
- Diana Baklizky
- Sue Koolmanojwong
- JoAnn Lane
- Stephany Ballomo
- Alejandro Bianchi
- Bob McCann
- Bob Epps

Workshop Agenda

- *Management of Technical Debt- Steve McConnell*
- *Technical Debt Observations- Jim Highsmith*
- *Types of Debt- Chris Sterling*
- *Workshop Exercise # 1-Identifying Technical Debt*
- *Management of Architectural Debt- Ipek Ozkapa*
- *Workshop Exercise # 2- Architecture Technical Debt*
- *Workshop Exercise # 3- System Design Technical Debt*
- *Operations & Maintenance(O&M) Technical Debt*
- *Workshop Summary/Action Plan*

When is it Technical Debt?

- Discussion converged on the following:
  - 3 I's – Intention, Impact, and Interest
  - Results from intentional decisions
    - Commits the program in a certain direction
    - Decision trade space has specific future impacts that are understood
    - Impacts have "interest payments" associated
    - Deferred decisions are only Technical Debt when the deferral causes accrued interest
  - Measurable – can identify and assess
  - Manageable – within the control of the program team

Difference of System and Software

- Software Technical Debt focuses more on:
  - Software product quality
- System Technical Debt includes:
  - System definition quality (requirements, architecture, interfaces, trades, analysis)
  - System operational/performance issues – both internal and interoperability
  - System life management issues (e.g., adaptability, extensibility, …)
  - System maintainability/sustainment issues

Other key points

- *What can create greatest impact?*
  - *Top level architectural issues*
    - *Dependency analysis may help highlight*
    - *Consideration of future needs and interoperability*
  - *Top system definition that touches or drives the most lower level elements*
- *Technical debt was not viewed as being a factor of System Complexity*
- *Identified additional boundaries for System Technical Debt, examples include:*
  - *Use of NDI is not necessarily TD*
  - *Expected iteration between processes is not TD*

Assessing the Total System Technical Debt?

- Technical debt is composed of impact costs from various categories of effort
  - Rework
  - Quality impacts other than rework
  - Performance impact
  - Maintainability/sustainment impacts
  - Adaptability/flexibility/evolution impacts (e.g., cost of change)
  - …
- More analysis needed to fully identify and define this set
- The total TD is the aggregate of these elements

Technical Debt and Risk

- *Technical debt involves both risk management and problem management*
- *Need to investigate more with Risk Management experts*
- *After decisions are made, TD has occurred and problem management is needed*
  - *Catalog and prioritize to pay down as ROI shows it is advantageous*

Next steps

- *Identify and define measures for System Technical Debt*
- *Create draft white paper*