A General Systems-Based Ecological Model of Information Technology

Phil Teplitzky

H P Marin & Touro College

Agenda

- Introduction
- Definition of Terms
- Life Cycles
- How the Lifecycles Interact
- Summary

Introduction

Teplitzky's Rules for Presentations – ***the key questions you should always ask yourself when listening to a presentation!***

1. What is going to be discussed?
2. Why are the objectives of the presentation?
3. Why am I interested in what is being presented?
4. How will it impact or change my life?

You should ALWAYS be asking yourself these questions when at a presentation, or a meeting in fact it is good advice for almost all occasions!

What you will learn – *if you follow Teplitzky's Rules*

- Software development is a team effort that requires the interaction of ***many different life cycles***
- It is an ***Ecosystem***, a failure in one component of an Ecosystem can cause the failure of the whole system – *there have been many examples of this throughout history*
  - —The Sahara Forest
  - —The melting Ice Caps
  - —Desiccation of the American South West
- The multiple Life Cycles that make up the Software Ecosystem interact according to the ***Rules of General Systems Theory***

Life Cycles that Comprise the Software Ecosystem

1. **Application Development** – the topics covered in the standard SDLC.
2. **Data and Data Base Management** – A data-centric SDLC.
3. **Quality Engineering / Testing** – the process of Functional, Stress and Integration testing. It would also include software metrics, for example, the Halstead and McCabe complexity model.
4. **People** – the processes of hiring, and managing technical and support people, or what may colloquially be called Human Resources (HR)
5. **Kulture** – the culture of the organization
6. **Training –** related to the maintenance and enhancement of skills and capabilities.
7. **Hardware** – the operation, maintenance and enhancement of the physical devices.
8. **Operations** – the execution of job streams and execution of jobs in the proper sequence.
9. **Networks** – everything related to WANs, LANs and both inter- and intra- communication for voice and data.

Life Cycles that Comprise the Software Ecosystem

10. **Audit and Control** – the establishment of an appropriate System of Internal Control and the periodic or continuous Audit of the System of Control.
11. **Security** – including both logical and physical authorization and authentication.
12. **Presentation Layer** – the design and instantiation of the presentation layer of the Applications, for example, the building of Web sites and portals
13. **Reporting** – the design, building and maintaining both regulatory and internal reports
14. **Business Intelligence & Analytics** – the design, build and maintenance of analytic capabilities, including SAS, System R / Sand other analytic engines.
15. **Deployment / Provisioning** – the process of rolling out and making available for General Use new Applications or Versions of Existing Applications
16. **Retirement / Replacement** – the processes associated with replacing old applications with new ones or merging multiple applications
17. **Data Governance** – this includes Meta Data Management, Master Data Management and creation of Data Dictionary and Directory Systems

Why learn about the Software Ecosystems?

- You will build better Applications and Data Structures
- Applications will be:
  - —**MODIFIABLE**
  - —**ENHANCEABLE**
  - —**TESTABLE**
  - —**SECURE**
- At **Lower TCO**, **Faster** and with overall higher **QUALITY**

Definition of Terms –the four that are relevant to this presentation, in fact there good to know in general!

1. System – *what are they*
2. System Life Cycles – *what makes them cyclical*
3. Ecosystem – *what is the difference between a System and an Ecosystem*
4. General System Models / Fuzzy Set Theory

What are your definitions of these TERMS?

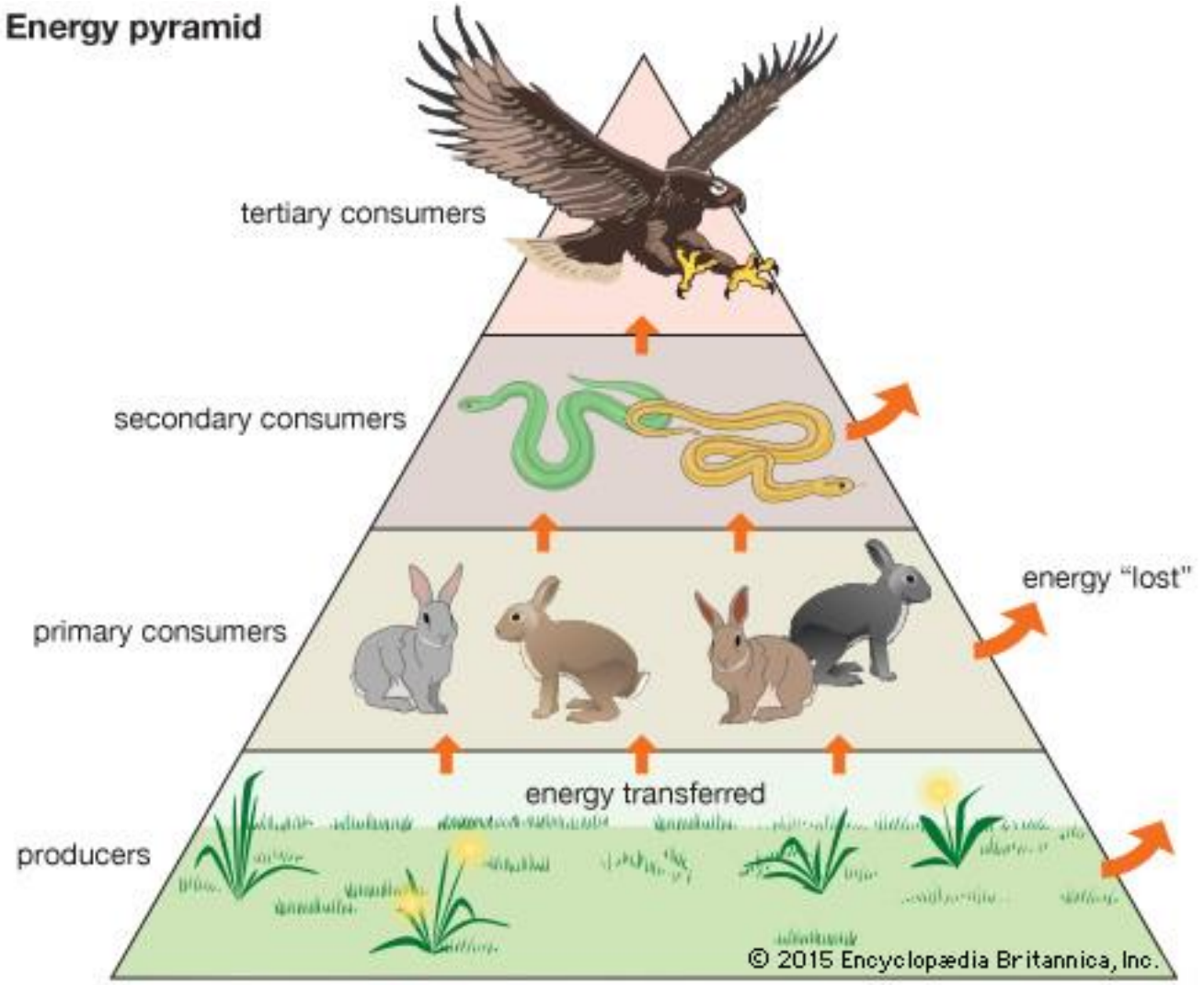
Systems Theory

- **System theory** is the transdisciplinary study of the abstract organization of phenomena, independent of their substance, type, or spatial or temporal scale of existence. It investigates both the principles common to all complex entities, and the (usually mathematical) models which can be used to describe them.
- **General Systems Theory** is about broadly applicable concepts and principles, as opposed to concepts and principles applicable to one domain of knowledge. It distinguishes dynamic or active **systems** from static or passive **systems**.

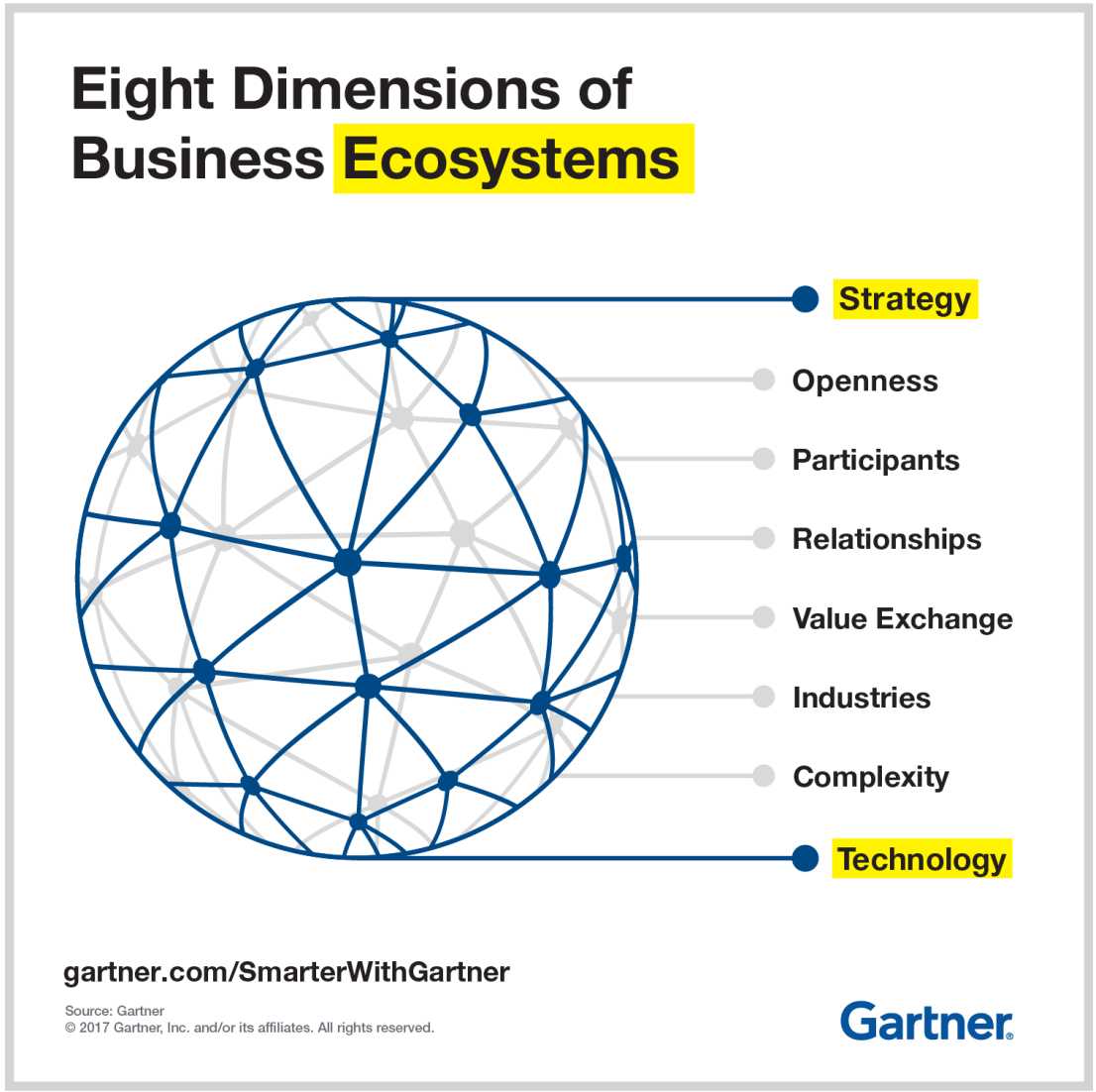
Ecosystem

- noun, ***Ecology***
- a system, or a group of interconnected elements, formed by the
  interaction of a community of organisms with their environm ent.
- any system or network of interconnecting and interacting par ts, as in
  a business: *The success of Apple's ecosystem depends on hardware/software integration.*
- *Manufacturers, retailers, and customers are all part of the au tomotive Industry's ecosystem.*

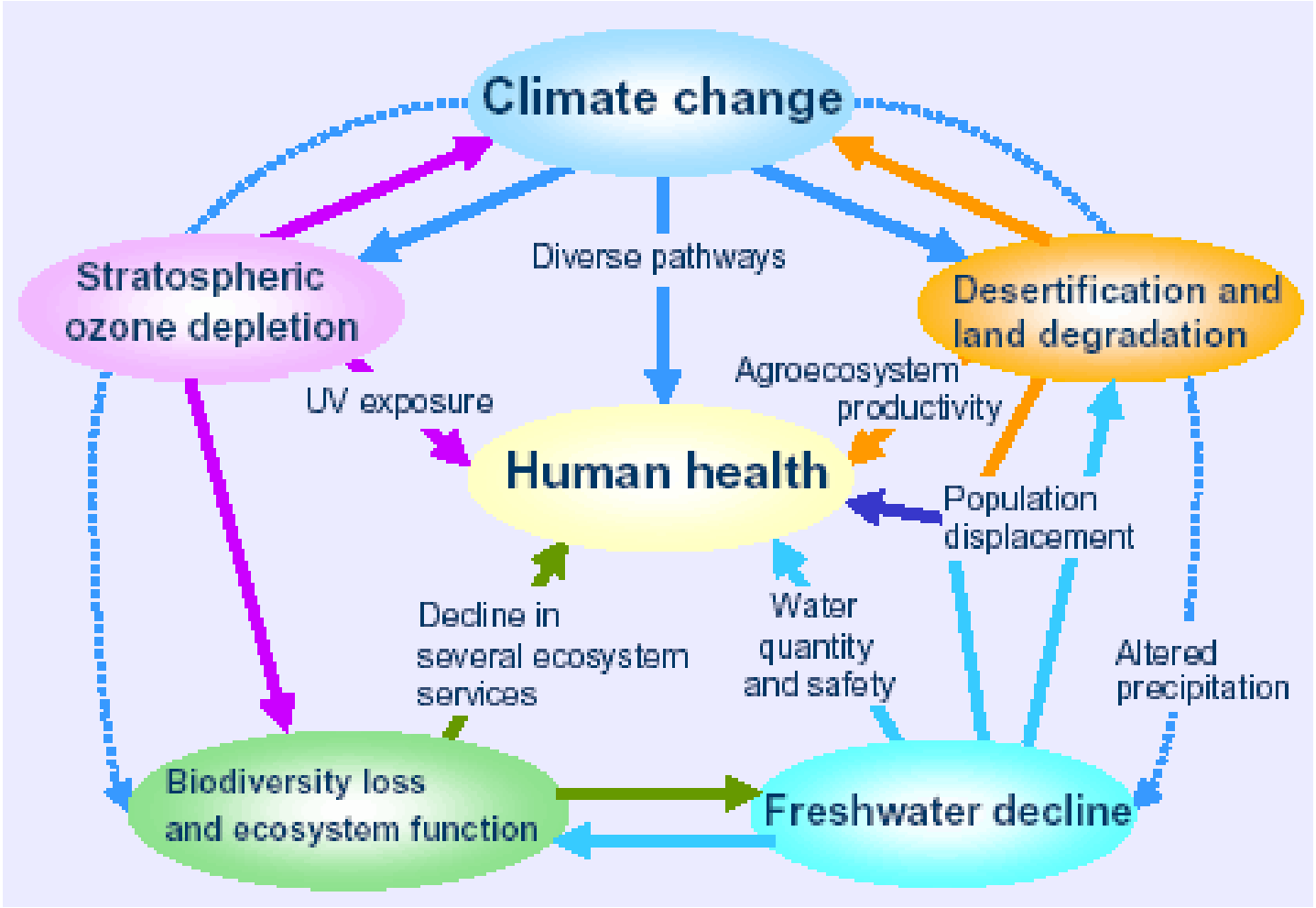
Ecosystem


Energy pyramid
tertiary consumers
secondary consumers
primary consumers
producers
energy transferred
energy "lost"
© 2015 Encyclopædia Britannica, Inc.

Ecosystem


Eight Dimensions of Business Ecosystems
Strategy
Openness
Participants
Relationships
Value Exchange
Industries
Complexity
Technology
gartner.com/SmarterWithGartner
Source: Gartner
© 2017 Gartner, Inc. and/or its affiliates. All rights reserved.
Gartner

Ecosystems


Climate change
Stratospheric ozone depletion
Diverse pathways
Desertification and land degradation
UV exposure
Agroecosystem productivity
Human health
Population displacement
Decline in several ecosystem services
Water quantity and safety
Altered precipitation
Biodiversity loss and ecosystem function
Freshwater decline

General Systems Theory

- General system theory, therefore, is a general science of "wholeness...The meaning of the somewhat mystical expression, "The whole is more than the sum of its parts" is simply that constitutive characteristics are not explainable from the characteristics of the isolated parts. The characteristics of the complex, therefore, appear as "new" or "emergent"...
- Compared to the analytical procedure of classical science with resolution into component elements and one-way or linear causality as a basic category, the investigation of organized wholes of many variables requires new categories of interaction, transaction, organization, teleology..."

Fuzzy Set Theory

- Fuzzy set theory was introduced by Zadeh in (1965) as another approach to represent uncertainty. A fuzzy set A is characterized by its membership function. This is a function whose range is contained in the unit interval. At a point the value of this function represents the degree of membership of the point in the set A.
- You might answer somewhere between 20 and 50 inches. A better answer could be the following fuzzy membership function. 20 25 30 35 40 45 50

In this case we are combining Fuzzy Set Theory and General systems Theory to describe the interaction between the Sixteen Life Cycles

Fuzzy Logic

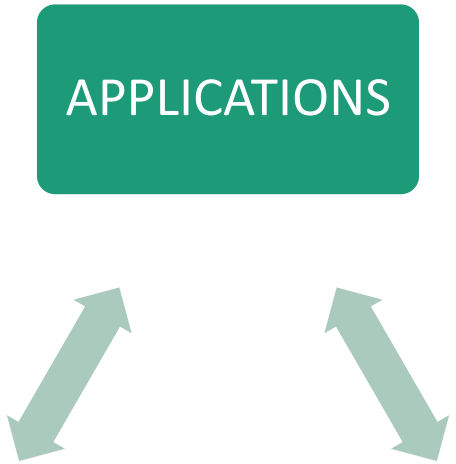
- Unlike two-valued Boolean logic, fuzzy logic is multi-valued. It deals with **degrees of membership** and degrees of truth.
- Fuzzy logic is not logic that is fuzzy, but logic that is used to describe fuzziness. Fuzzy logic is the theory of fuzzy sets, sets that calibrate vagueness
- Unlike two-valued Boolean logic, fuzzy logic is multi-valued. It deals with degrees of membership and degrees of truth.
- In fuzzy set theory: an element is with a certain degree of membership. Thus, a proposition is not either true or false, but may be partly true (or partly false) to any degree. This degree is usually taken as a real number in the interval [0,1].

In this case the level of interaction between the Ecosystems Components is Fuzzy dependent upon the Type of Ecosystem

Life Cycles of Relevance

1. **Application Development** – the topics covered in the standard SDLC.
2. **Data and Data Base Management** – A data-centric SDLC.
3. **Quality Engineering / Testing** – the process of Functional, Stress and Integration testing. It would also include software metrics, for example, the Halstead and McCabe complexity model.
4. **People** – the processes of hiring, and managing technical and support people, or what may colloquially be called Human Resources (HR).
5. **Kulture – Culture**
6. **Training –** related to the maintenance and enhancement of skills and capabilities.
7. **Hardware** – the operation, maintenance and enhancement of the physical devices.
8. **Operations** – the execution of job streams and execution of jobs in the proper sequence.
9. **Networks** – everything related to WANs, LANs and both inter- and intra- communication for voice and data.
10. **Audit and Control** – the establishment of an appropriate System of Internal Control and the periodic or continuous Audit of the System of Control.
11. **Security** – including both logical and physical authorization and authentication.
12. **Presentation Layer** – the design and instantiation of the presentation layer of the Applications, for example, the building of Web sites and portals
13. **Reporting** – the design, building and maintaining both regulatory and internal reports
14. **Analytics** – the design, build and maintenance of analytic capabilities, including SAS, System R / Sand other analytic engines.
15. **Deployment / Provisioning** – the process of rolling out and making available for General Use new Applications or Versions of Existing Applications
16. **Retirement / Replacement** – the processes associated with replacing old applications with new ones or merging multiple applications
17. **Data Governance** – including Master Data Management, Business Data Glossary and Data Dictionary / Directory

Life Cycle Interaction Model



The interaction between the Life Cycles varies by the type of Ecosystem – the level of Interaction, the impact can vary from 1 to .1

The interaction between Applications and Data will vary depending upon:

- **Use of a DBMS**
- **Flat Files**
- **Data Sharing**

The interaction between Data and Network will be dependent upon the level of Data Distribution and between Applications and Network is dependent upon the Applications being Scheduled or Transactional, and Volume

How Do The Life Cycles Interact?

- Is it Boolean?
- Or is it a Fuzzy or General System Model?

Does the level of interaction the degree of Fuzziness dependent upon the factors we have identified?

Life Cycles – *Factors that Impact Interactions*

- How do the Life Cycles interact with each other?
- Does it matter if you are doing:
  - —New development
  - —Enhancements
  - —Modifications
  - —Adapting to meet increased Transactions / Volumes
- Are large systems the same as small systems, and how they impact the Life Cycle interactions?

Life Cycles – *Factors that Impact Interactions*

- Does the Software Development methodology being used matter?
  - —Waterfall
  - —Agile
  - —Structured
  - —GTVX
- How does a Transactional System vs a Scheduled (Batch) system change the interactions of the Life Cycles
- Distributed vs Centralized vs Cloud
- Flat files vs Centralized Data Base (DBMS) vs Distributed Data Base – serial update vs Multiphase commit?
- High level of Data Sharing vs low Data Sharing

Life Cycles – *Factors that Impact Interactions*

- High volume vs low volume
- Constrained Processing Windows – *must be done overnight*
- Level of Security – *low to National*
- Reliability – *what is the acceptable MTTF, the lower the MTTF the costlier the Ecosystem*
- Staff Capability in the Ecosystem – *how experienced and how capable, you can have a lot of experience but low capability, experience is not measured in years!*
- Kulture

Can you think of other factors?

Life Cycles – *Factors that Impact Interactions*

- These factors are Orthogonal to each other
- A change in anyone of them can impact the degree of the relationships between the Life Cycles
- A change in one Factor can impact other Factors
- Often more than one Factor may be changed at the same time – in that case you must determine the degree of interaction and their impact on each other and then there collective impact on the Life Cycles

In essence you have two General Systems, Life Cycles and Factors, Orthogonal to each other and each has a Fuzzy Logic relationship to its components

Our Opinion

- Complex interactions
- Follow General Systems Model rules
- The interaction is Fuzzy not Boolean
- All the factors interact and influence the type of interactions
- Every Company and every project have a different MODEL – and the model changes as you gain experience

Summary

- You need to know what the state of your individual Company
- You have to know what you are doing – the Taxonomy of the project and how it impacts and effects your Model
- But it is dependent upon:
  - —You knowing your model
  - —Understanding how the components inter-act
  - —What you are going to do
  - —And be able to understand how the what will impacts what is!

Summary

- If you do not understand the inter-relationships between the components Life Cycles you will increase your probability of ***failure***
- Since it is an Ecosystem a *failure in one* component will result in a *failure in the Ecosystem – not enough water or nitrogen and the crops will not grow*
- Software is the same, a failure in one Life Cycle – *not enough Bandwidth, the response times will deteriorate, and reach a threshold where the Ecosystem is no longer viable*
- But the relationships between the Life Cycles, the level of membership varies by a large number of factors, that change over time. You need to build the model and update it to understand the dependencies and make rational decisions