Team Software Development Project- CIS7015

Session 1: Digital Innovation

Dilani Wickramaarachchi, PhD in Software Engineering
Senior Lecturer, Faculty of Science
University of Kelaniya

At the Session…..

- **Look** - at what is being shown
- **Listen** - to what is being said
- **Think** - about the examples
- **Ask** - questions when you do not understand
- **Write** - when you think it is important
- **Involve-** actively involve for group discussion

Overview

- Introduction to the Module
- Discuss evaluation criteria
- Digital Innovation and related terms
  - Digital transformation
  - Digitization and digitalization
  - Invention and innovation
- How to proceed?
- Basics about Software…
  - Software vs Hardware
  - Software Development Life Cycle
  - Software Engineering
  - Software Applications and its Evolution
  - Information flow in Software Projects
  - Software Project Success- Standish Group Chaos Report

Team Software Development Project

- **Aim(s)**
  - This module aims to provide students with a practical and reflective experience of developing a software prototype in a team.

- **Learning Outcomes**
  - On successful completion of the module students should be able to:
    - Critically evaluate and manage self and peer learning, progress and performance effectively
    - Analyse issues involved in collaborative software development
    - Justify selection of appropriate methods, tools and techniques to develop a prototype

Team Software Development Project

Learning and Teaching Delivery Methods		
Lectures/Tutorials/Workshops	48 hours	/36 hours
Student Centred Learning	152 hours	
Total	200 hours	

Assessment Methods			
Assessment Type	Duration/Length of Assessment Type	Weighting of Assessment	Approximate Date of Submission
Proposal Submission Interim Presentation PRES1 - Group Presentation	1500 words equivalent	5% 20%	Week 3 Week 6 Week 12
WRIT1 - Group Report	2100 words equivalent	35%	Week 12
WRIT2 - Individual Report	2400 words equivalent	40%	Week 12

Team Software Development Project

- **Indicative Content**
  - Overview of Software Engineering Process, Lifecycle Models and Stages
  - Team versus Individual Projects
  - Team Skills and Structures
  - Team Roles and Role Allocation
  - Team Management Issues
  - Reasons for prototyping
  - Strategies for Prototyping
  - Characteristics of prototypes
- **Recommended Reading & Required Reading**

- Roger S. Pressman, Software Engineering: A Practitioner's Approach (7th Edition), Harlow: McGraw-Hill, 2009
- Ian Sommerville, Software Engineering (9th Edition), Harlow: Pearson, 2010
- Olga Filipova, Software Development from A to Z: A Deep Dive Into All the Roles Involved in the Creation of Software, (1st Edition), Apress, 2018
- Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer, (1st Edition), Apress, 2018

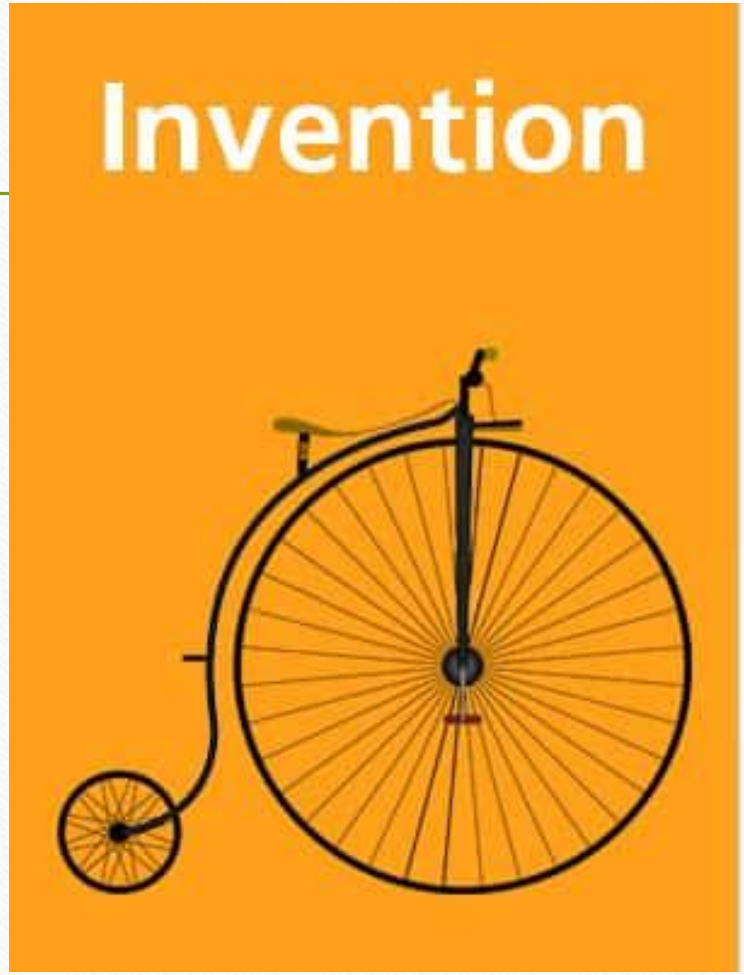
Student Centered Learning

- You hear you forget
- You see you remember
- You do you understand

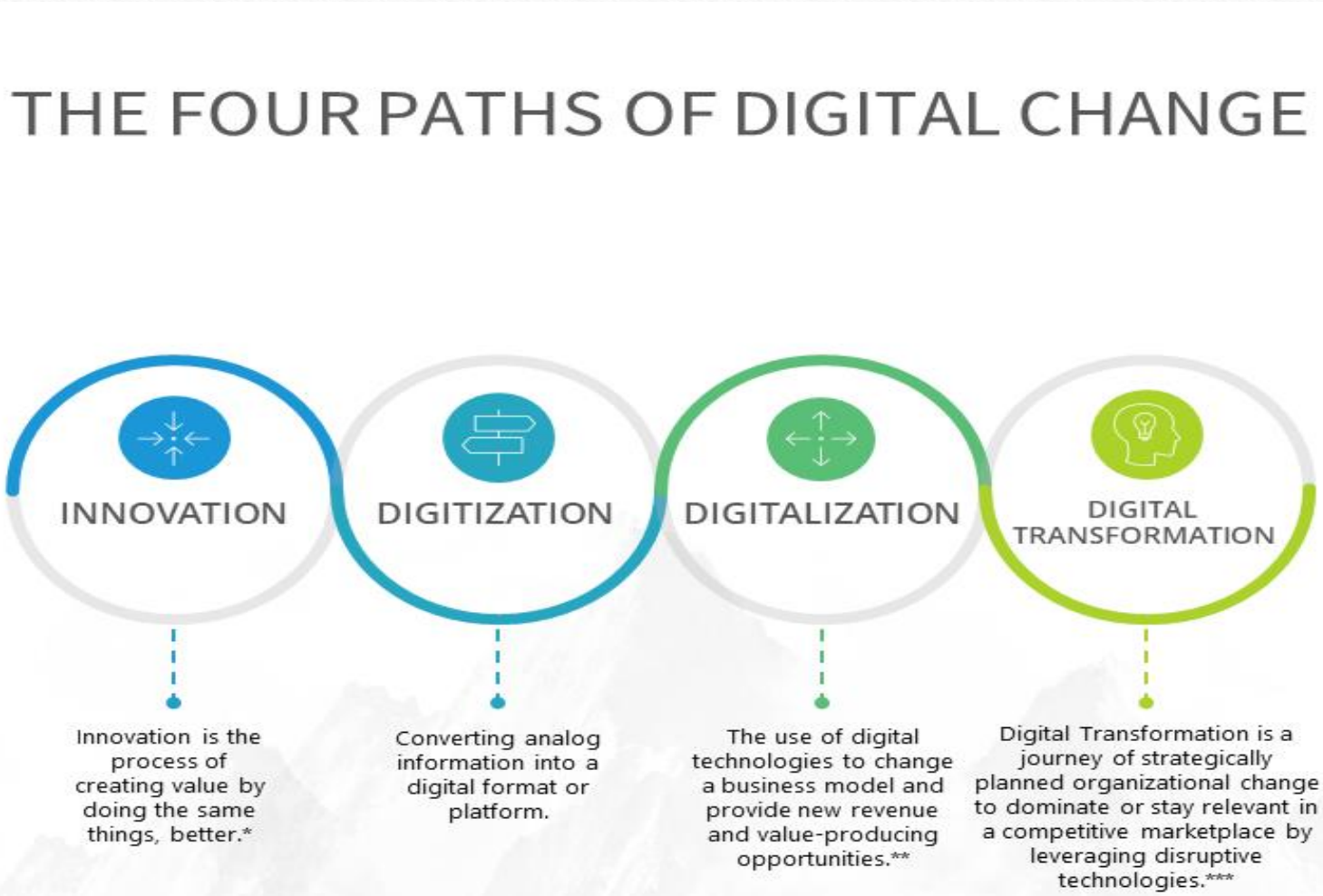
Useful Terms….

- Literacy, IT Literacy, Data Literacy, Digital Dexterity
- Invention, Innovation
- Digitization, Digitalization, Digital Transformation

Invention vs Innovation



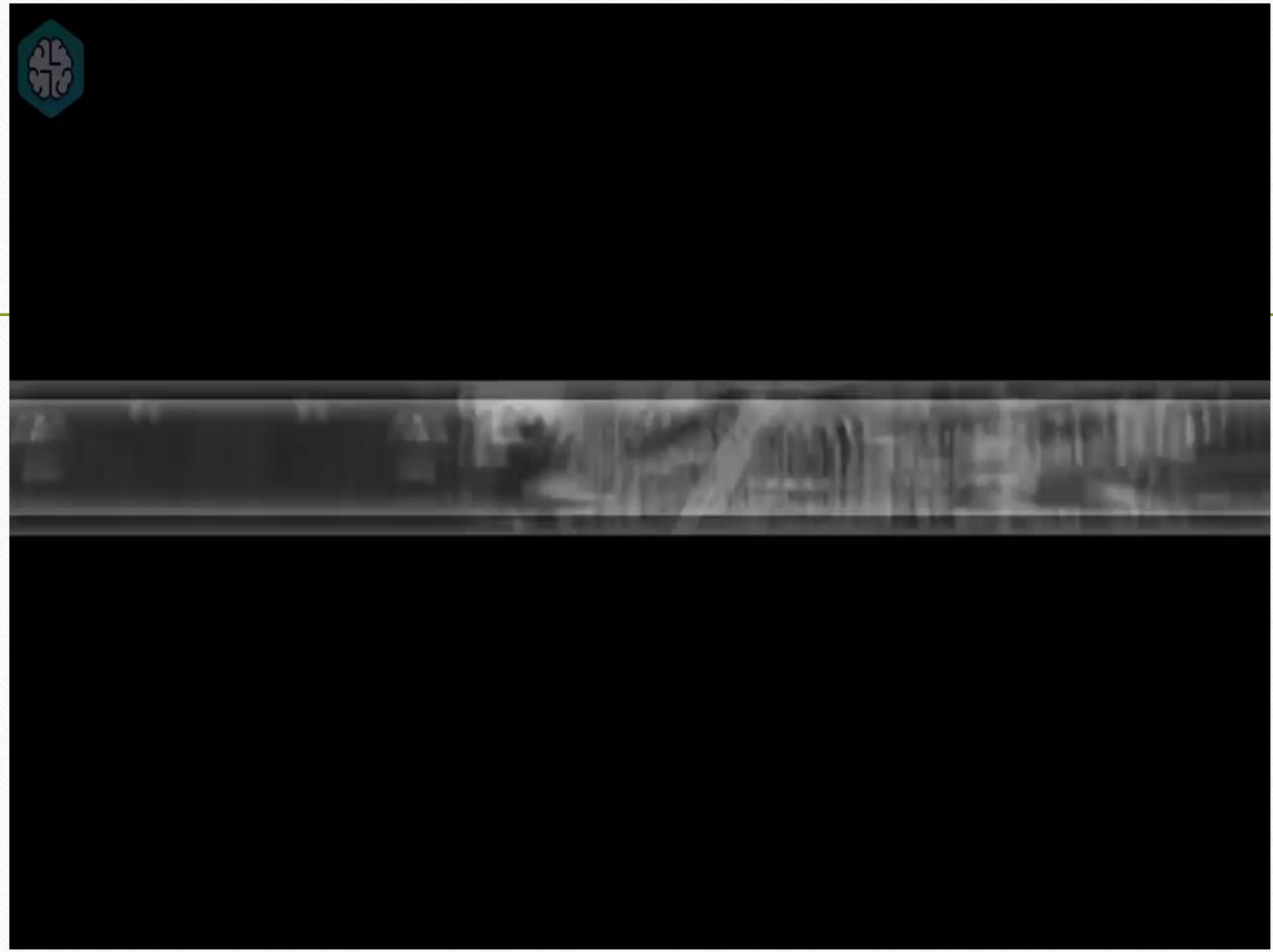
Digital Transformation



Digital Transformation

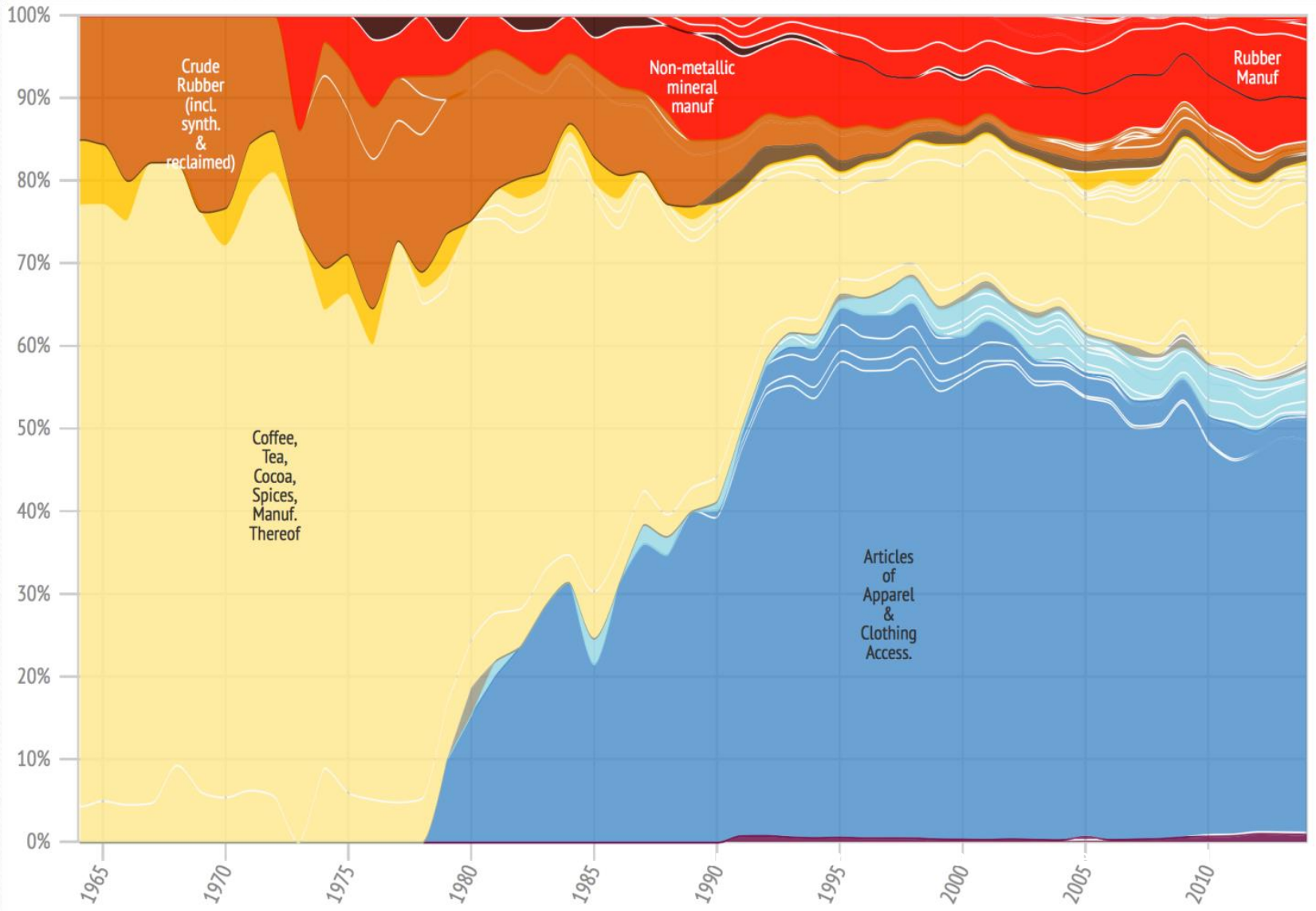
- Digital transformation is the process of using digital technologies to create new or modify existing business processes, culture, and customer experiences to meet changing business and market requirements. This reimagining of business in the digital age is digital transformation. *-Salesforce-*

- Digital Transformation is
  - the adoption of digital technology
  - to transform services or businesses,
  - through replacing non-digital or manual processes with digital processes or replacing older digital technology with newer digital technology. Wikipedia

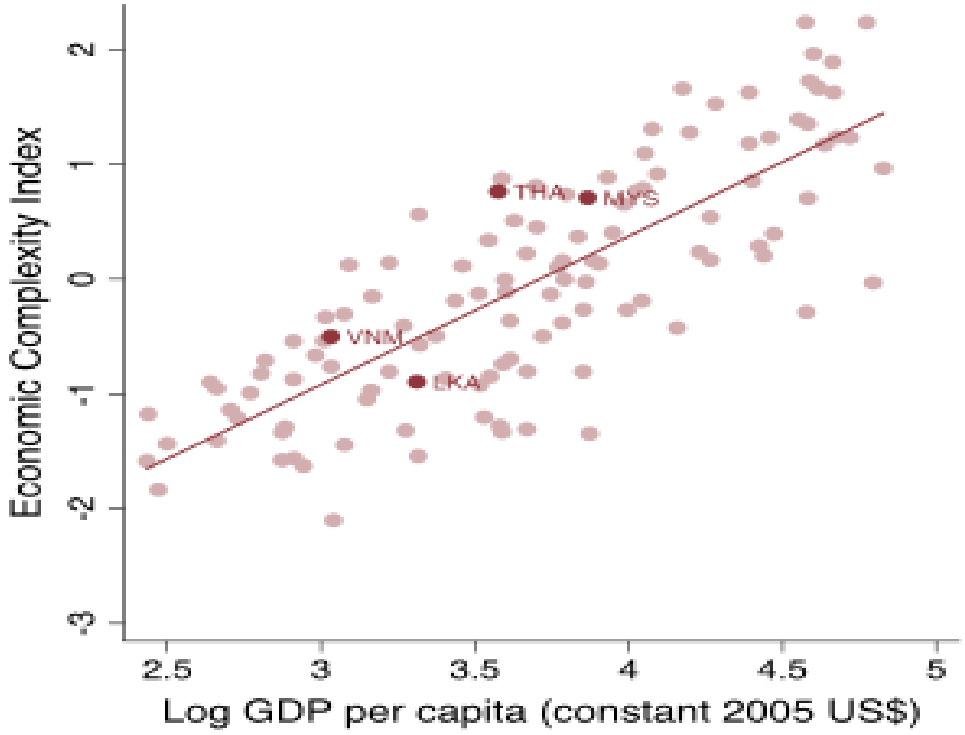


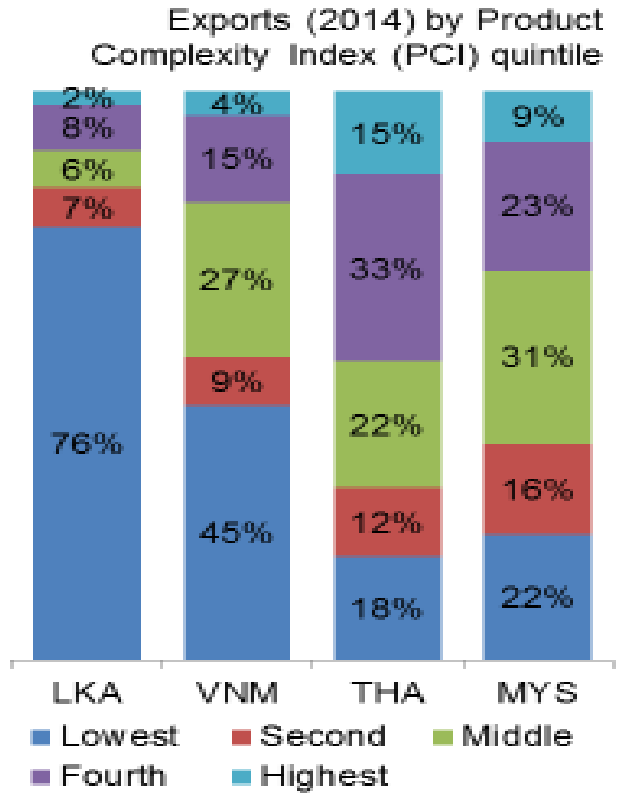
Grand Global Challenges

Source: Center for International Development, Harvard University



Likewise, Sri Lanka's exports are biased towards low-complexity products, implying low know-how.





GDP and ECI data from 2014. Source: calculations based on data from COMTRADE and WDI.

Source: Center for International Development, Harvard University

How do you start?

- A good question…that is 'worthy' of looking at
- Does not have to be a 'complex' one
  - simple and **DEEP**
- Look around and find problems…don't create new problems
  - A problem, but NOT a status report
  - Write about 1-page in your **OWN words**
  - **NO literature** here… simply stating the problem
  - After that, you can search literature and revise your idea..

Theory and Emerging Technology

- After you got the problem, you need to think of a feasible solution/s.
- Then you need to FIND the way to develop the proposed solution
  - You need to use theoretical knowledge you have gained
  - Need to look for emerging technologies
  - Then, using suitable approach need to implement the solution

Software

(1) instructions (computer programs): when executed provide desired features, function, and performance;

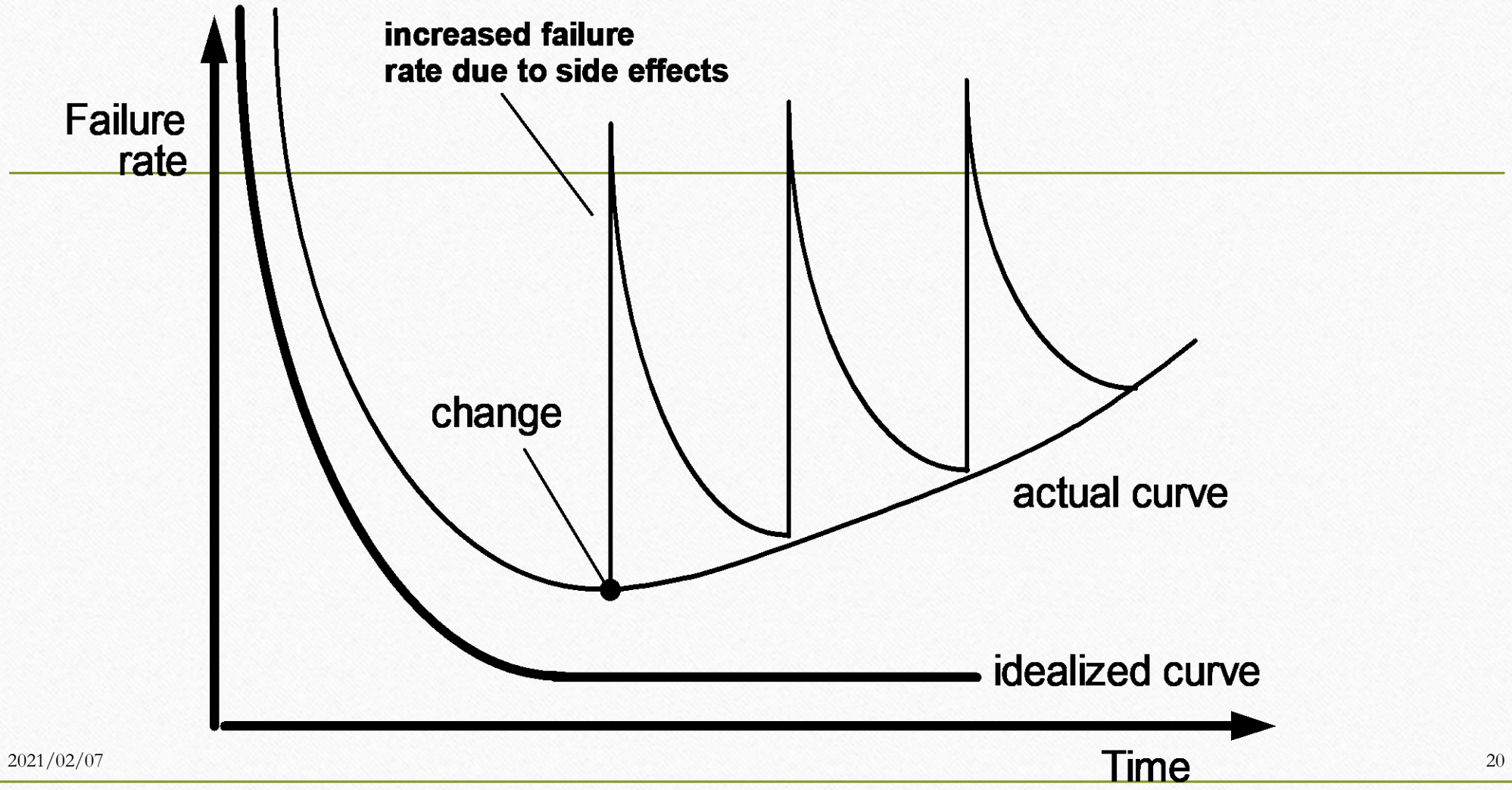
(2) data structure: enable the programs to adequately manipulate information

(3) documentation: describes the operation and use of the programs

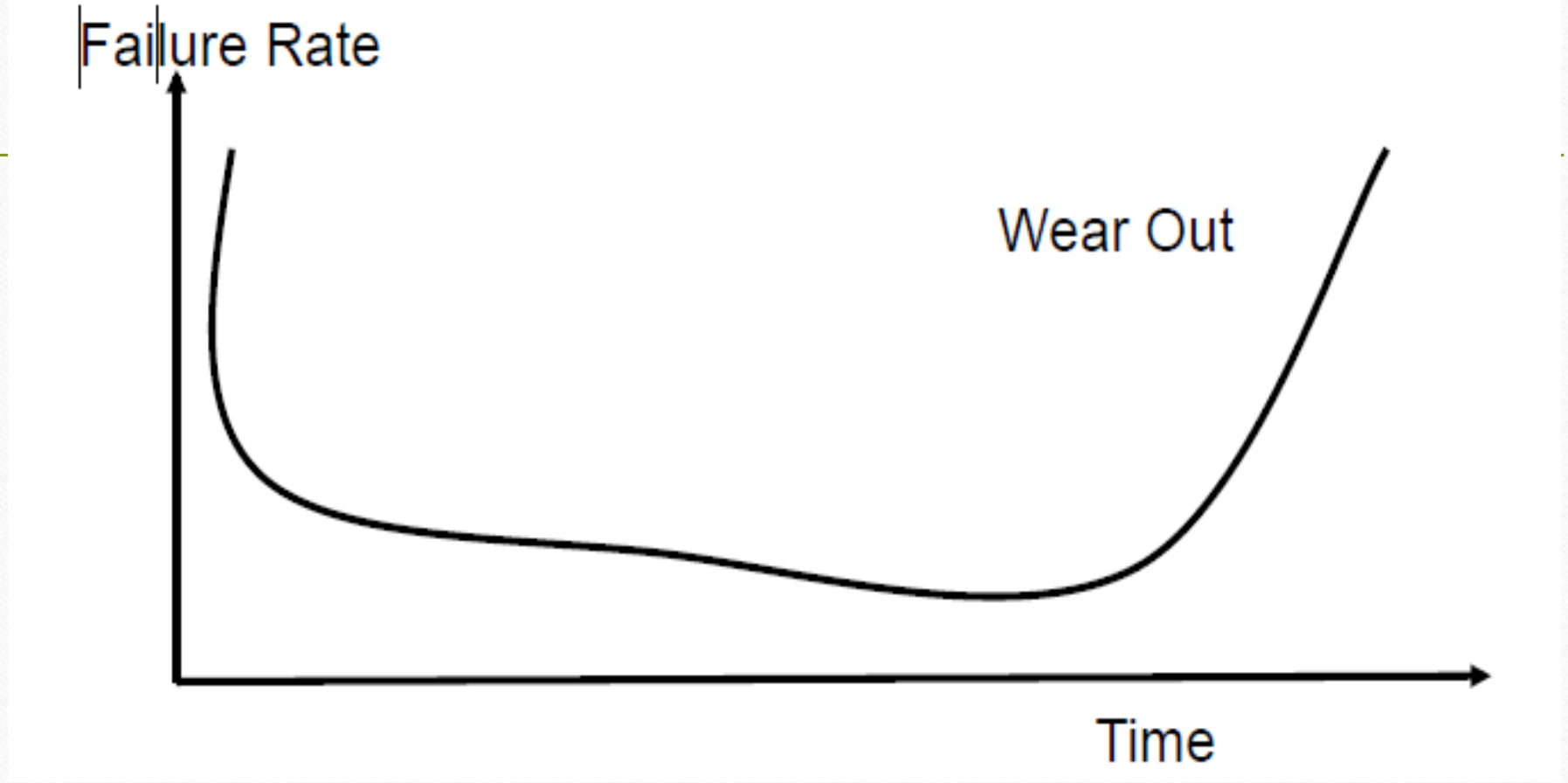
Software vs Hardware

- Software – how it differs..
  - Software is developed or engineered, it is not manufactured in the classical sense.
  - Software doesn't "wear out."
  - Labour requirement

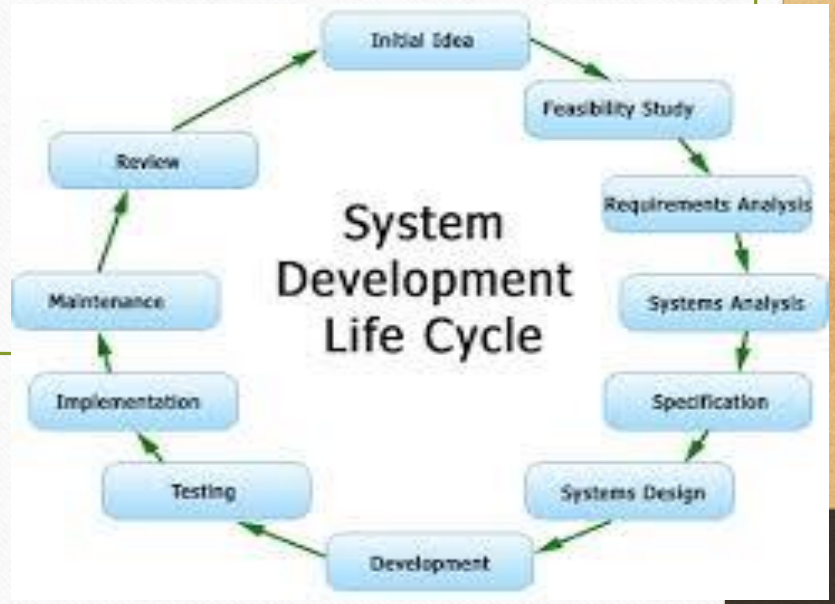
Software-Failure Rate vs Time



Hardware-Failure Rate vs Time



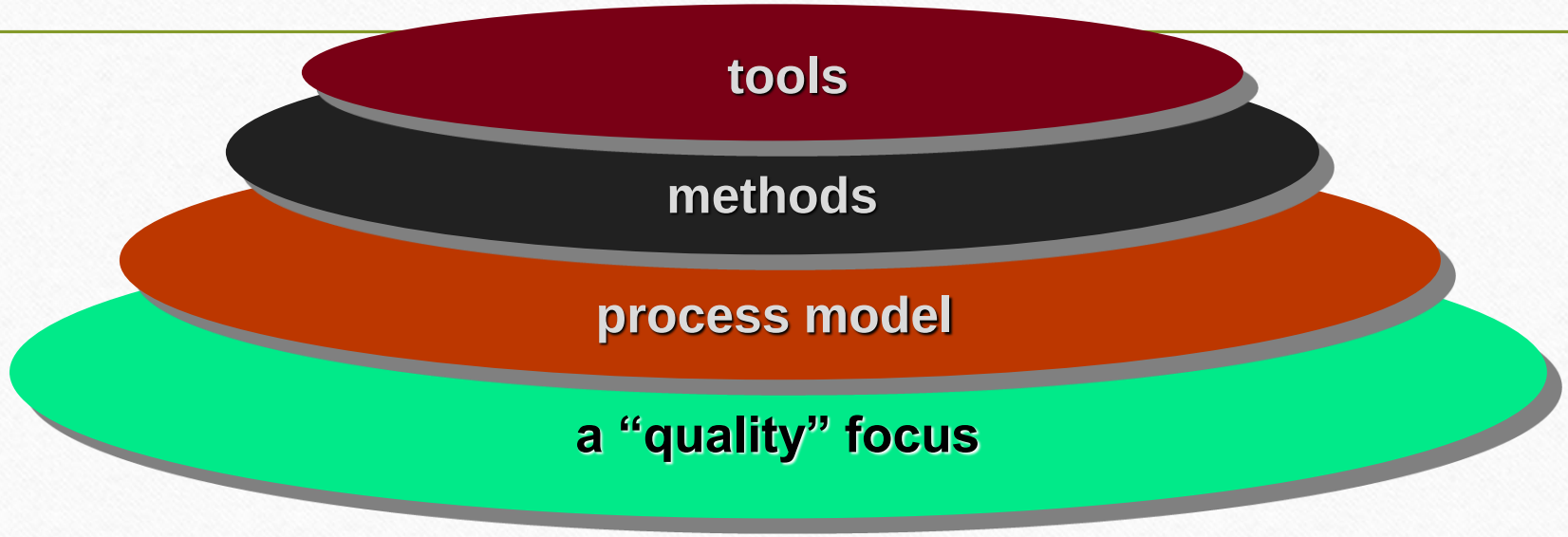
Software Development life Cycle (SDLC)



Software Engineering

- The seminal definition:
  - [Software engineering is] the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.
- The IEEE definition:
  - *The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*

Software Engineering: A Layered Technology



Software Engineering

Software Applications

- System software
- Application software
- Engineering/scientific software
- Embedded software
- Product-line software
- Web applications
- BI software and Big Data
- IoT

Software-should it change?.. Legacy to IoT

- software must be **adapted** to meet the needs of ***new computing environments or technology.***
- software must be **enhanced** to implement ***new business requirements.***
- software must be **extended to make it interoperable** with other more ***modern systems or databases***.
- software must be **re-architected** to make it viable within a network environment.

Information flow in SW projects

How the customer explained it

How the Project Leader understood it

How the Software architect designed it

How the Programmer wrote it

How theTester left it

How the Business Consultant described it

How the project was documented

How the customer was billed

How it was supported

What the customer really needed

Source: unknown

Task

- What do you mean by success of a software project?

Standish Group Chaos Report

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015	2020
SUCCESSFUL	29%	27%	31%	28%	29%	31
CHALLENGED	49%	56%	50%	55%	52%	50
FAILED	22%	17%	19%	17%	19%	19

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Success of a Software Project

- Project Management Institute (PMI) has defined success as onTime, onBudget, and onTarget also known as the *Triple Constraints*
- Six factors now being included in the overall measure of success
  - on Time, On Budget, on Target, on Goal, Value and Satisfaction

Bad SE practices create ..

- Failed projects
- Lost money
- Stressed employees
- Poor customer value

YEAR	COMPANY	OUTCOME (COSTS IN US $)
2005	Hudson Bay Co. [Canada]	Problems with inventory system contribute to $33.3 million* loss.
2004–05	UK Inland Revenue	Software errors contribute to $3.45 billion* tax-credit overpayment.
2004	Avis Europe PLC [UK]	Enterprise resource planning (ERP) system canceled after $54.5 million† is spent.
2004	Ford Motor Co.	Purchasing system abandoned after deployment costing approximately $400 million.
2004	J Sainsbury PLC [UK]	Supply-chain management system abandoned after deployment costing $527 million.†
2004	Hewlett-Packard Co.	Problems with ERP system contribute to $160 million loss.

List of failed and overbudget custom software projects

https://en.wikipedia.org/wiki/List_of_failed_and_overbudget_custom_software_projects

2008	2013	Digital Media Initiative	Digital production, media asset management	United Kingdom	State broadcaster	By 2013, the project was judged to be obsolete (as much cheaper commercial off the shelf alternatives by then existed) and was scrapped by BBC management. The BBC Director General said it had been a huge waste of money.[6]	more than £98m *(£81.7m)*	Outsourced, then insourced, then outsourced again	Cancelled
2009	2013	The Surrey Integrated Reporting Enterprise Network (SIREN)	Crime & criminal intelligence logging system	United Kingdom (Surrey)	Police Force	Not fit for purpose[7]	£14.8m	Outsourced	Scrapped
2011	2014	Pust Siebel	Police case management	Sweden	Police	Poor functioning, inefficient in work environments.[8]	SEK 300m ($35m)[9]	Outsourced	Scrapped
2012	2014	Cover Oregon	Healthcare exchange website	United States	State government	Site was never able to accept online enrollments, so users were instructed to mail in paper enrollments instead.	approx $200m	Outsourced	Cancelled, then client and supplier both sued each other

Good SE practices create ..

- Successful projects
- Happy customers
- Business value
- Lower stress levels for developers

Main objective of this Module

- Use good software engineering practices and develop a successful software with your team.

End of the Session 1 -

Go back to Overview