

Software Development Context

Yong ZHANG
September, 2020





Contents

- Software Architecture
- CMMI and Agile Software Development
- The Systems View of Project Management
- Understanding Organizations
- Project Phases and Project Life Cycle
- The Context of IT Projects
- The Recent Trends of IT Projects

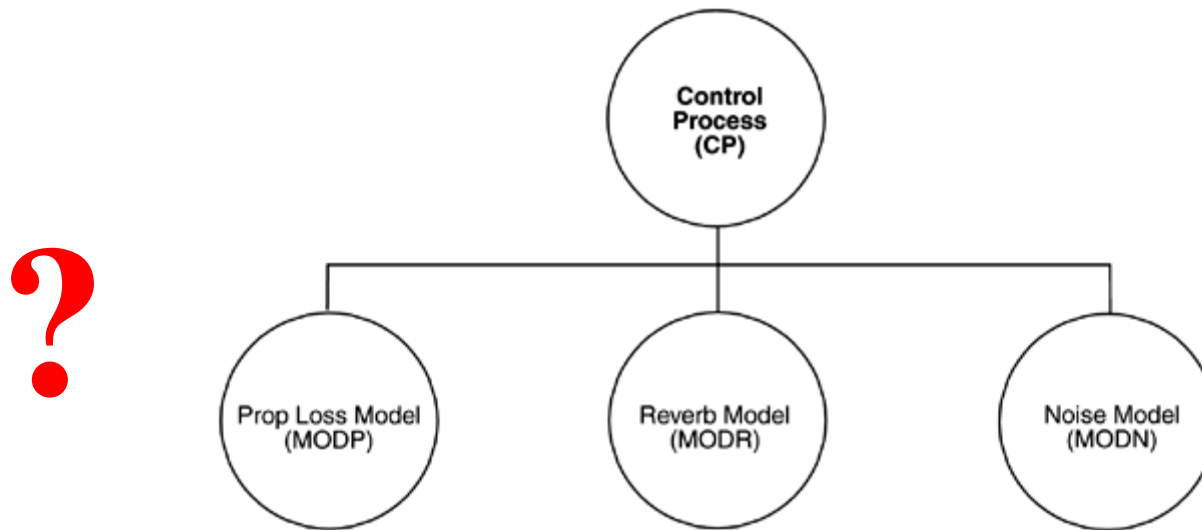
UNDERWATER SOUNDS

Deep Relaxation | Sleep | Meditation



What is software architecture ?

What is not software architecture ?

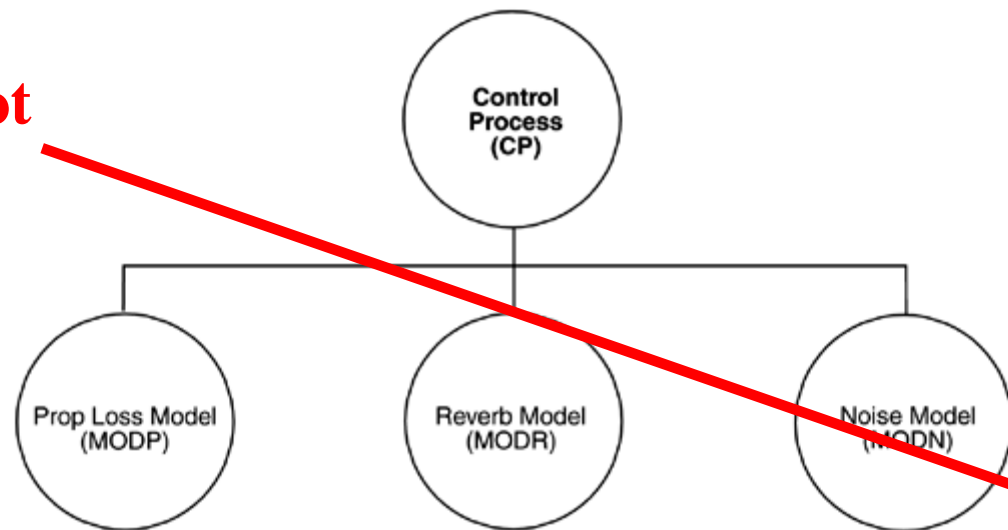


- Example: the figure claims to describe the architecture of a system that simulates subaquatic acoustics
- It uses a very popular informal notation - *boxes-and-lines*
- “*boxology*”

What is software architecture ? (cont)

- What can we **not learn** from this figure?
 - What is the type of the elements ?
 - What are the responsibilities of each element?
 - What is the significance of connections?
 - What is the significance of alignment?
- Conclusion: not a software architecture representation (or at least not a *useful* representation of a software architecture)

Architecture should not be described like this !



Software architecture - definition

- “The software architecture is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements and the relationships among them”

(Bass et al)



Software architecture definition explained (1)

“The software architecture is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements and the relationships among them”

(Bass et al)

- Software elements: (the term *of architectural element* is preferred instead of *architectural component*.)
 - Architecture is an **abstraction** of a system
 - Architecture is hiding certain details
 - Externally visible properties: the provided services, performance characteristics, shared resources, etc.

Software architecture definition explained (2)

“The software architecture is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements and the relationships among them”

(Bass et al)

- **Structures of the system**: Architecture is described by multiple different structures (views):
 - The modules structure - the static structure of a system, results in the design phase
 - The interactions structure – the dynamic structure that can be observed at runtime
 - The allocation/deployment structures – describes how software elements are mapped onto non-software elements (hardware, staff)



Software architecture definition explained (3)

“The software architecture is the structure or structures of the system, which comprise **software elements, the externally visible properties of those elements and the **relationships among them**”**

(Bass et al)

- Inside each structural view, software elements and relationships among them may have different meanings:
 - **Elements** can be: objects, classes, functions, processes, programs, libraries, databases, etc.
 - **Relationships** between elements: part of, synchronization, function call, etc.



Example

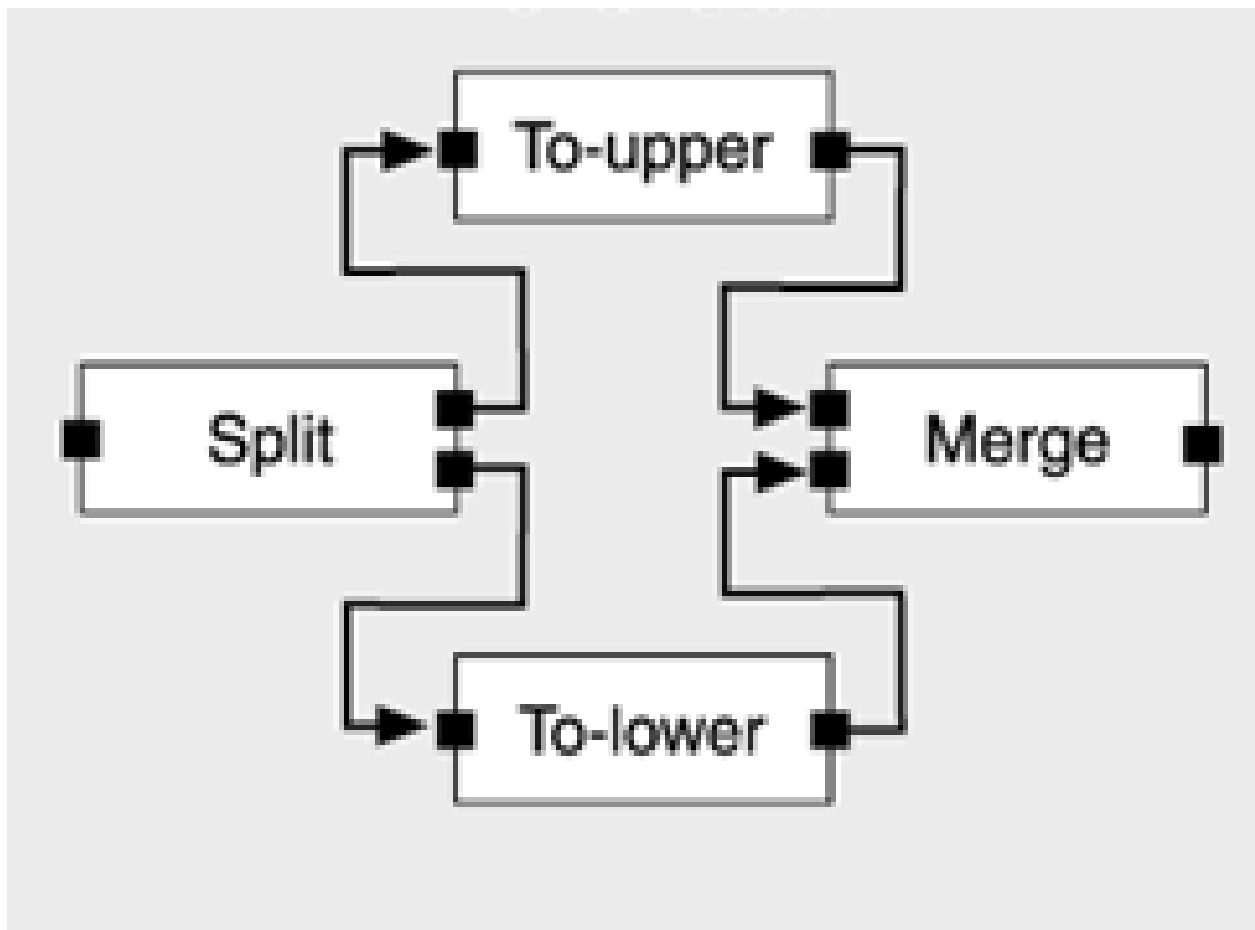
Describe the architecture of a system doing simple text processing: the lines of the original text are transformed in following way:

odd lines are rewritten in upper case and
even lines are rewritten in lower case.

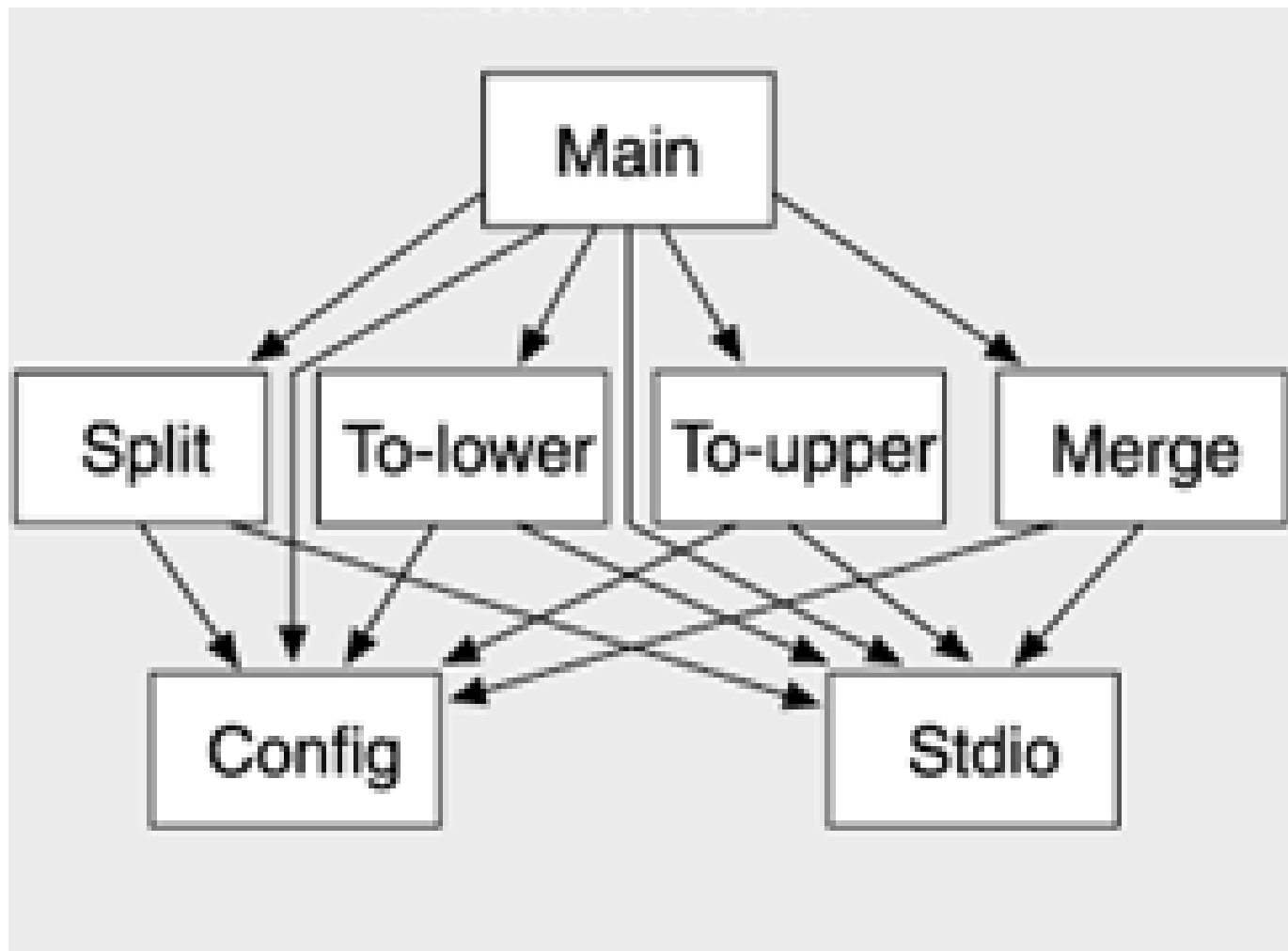
- Two different engineers draw following “boxology” diagrams:



Example – Diagram 1



Example – Diagram 2



Example – The Architecture

- No one of the 2 diagrams represents the software architecture of the system and neither a rigorous described structural view
- The software architecture of the system is composed by both views

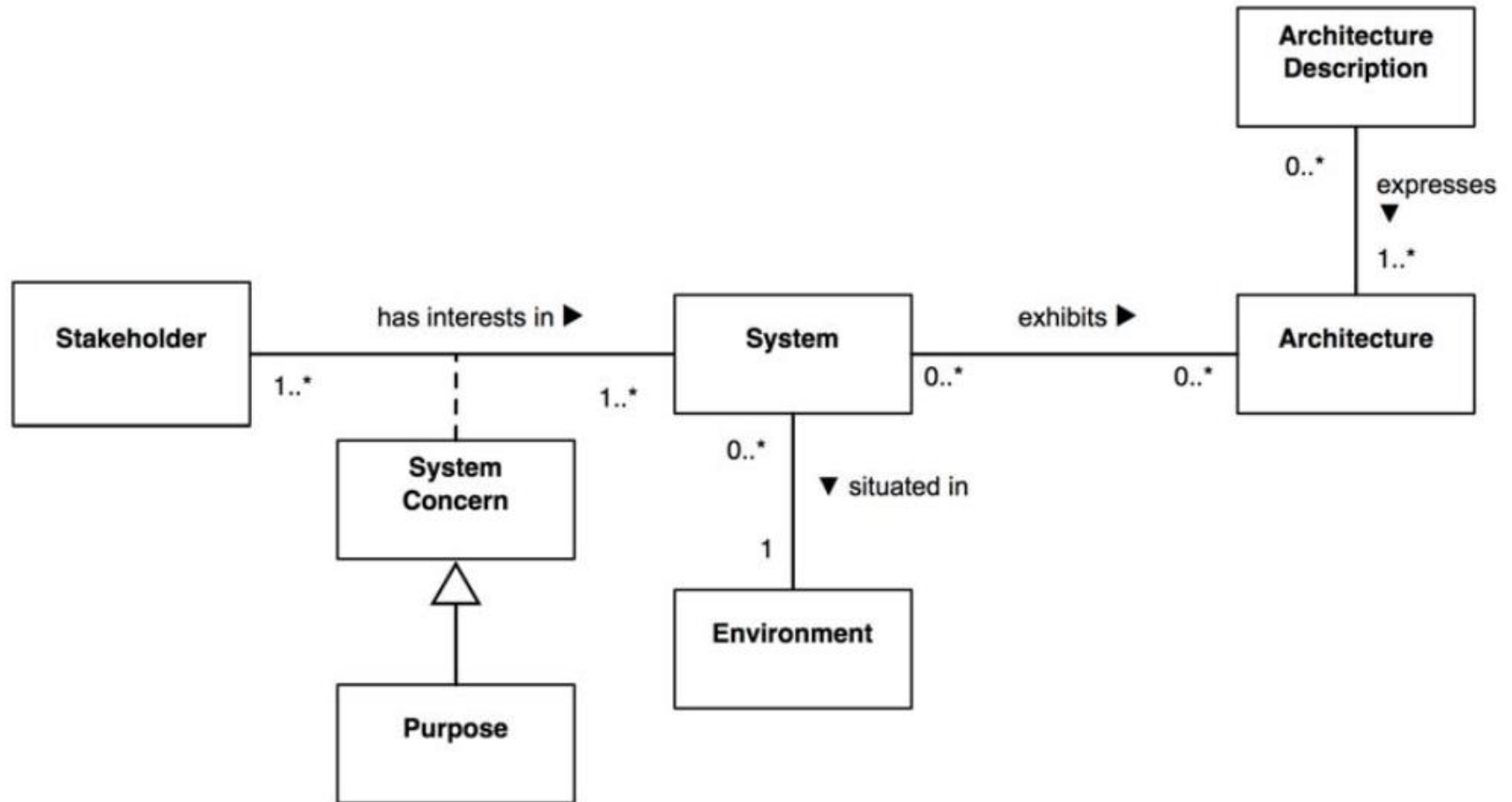


Describing software architectures

- **IEEE/IEC/ISO 42010:2011** “Systems and software engineering – Architecture description”
- Revises the earlier IEEE Std 1471:2000 “Recommended Practice for Architectural Description of Software-intensive Systems”
- It specifies the requirements for architectural descriptions
- It does not specify any description language
 - ADL’s (Architecture Description Languages): Wright, ACME, SysML, Rapide, etc
 - UML



Context of Architecture Description



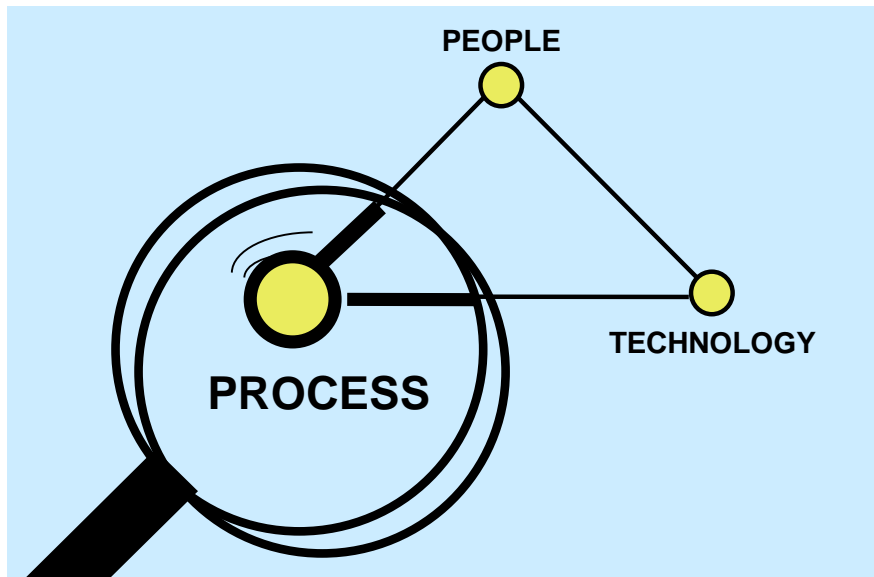


Contents

- Software Architecture
- CMMI and Agile Software Development
- The Systems View of Project Management
- Understanding Organizations
- Project Phases and Project Life Cycle
- The Context of IT Projects
- The Recent Trends of IT Projects

Quality Leverage Points

Everyone realizes the importance of having a motivated, quality work force but...



...even our finest people can't perform at their best when the process is not understood or operating "at its best."

Major determinants of product cost, schedule, and quality

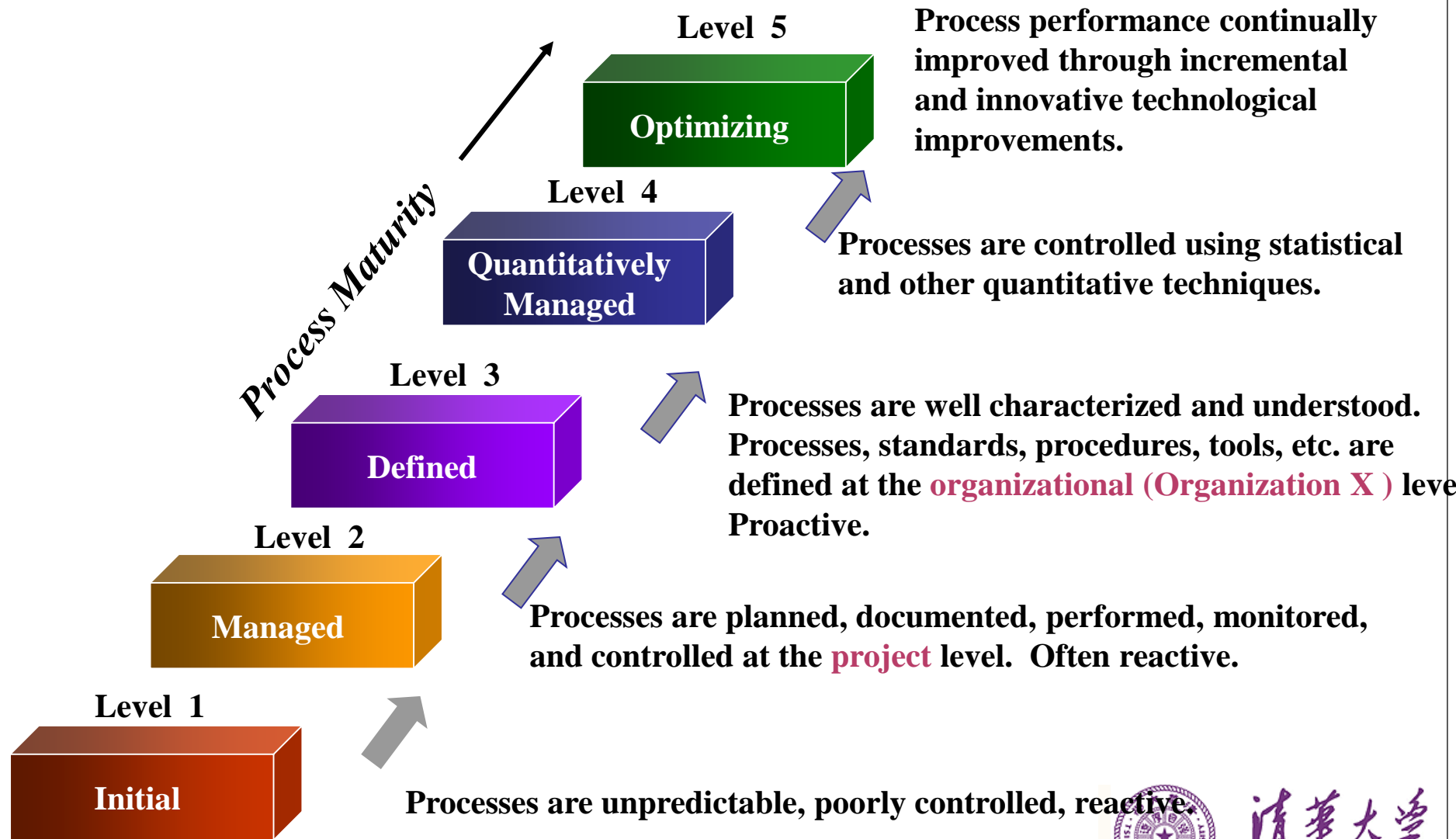


What is CMMI?

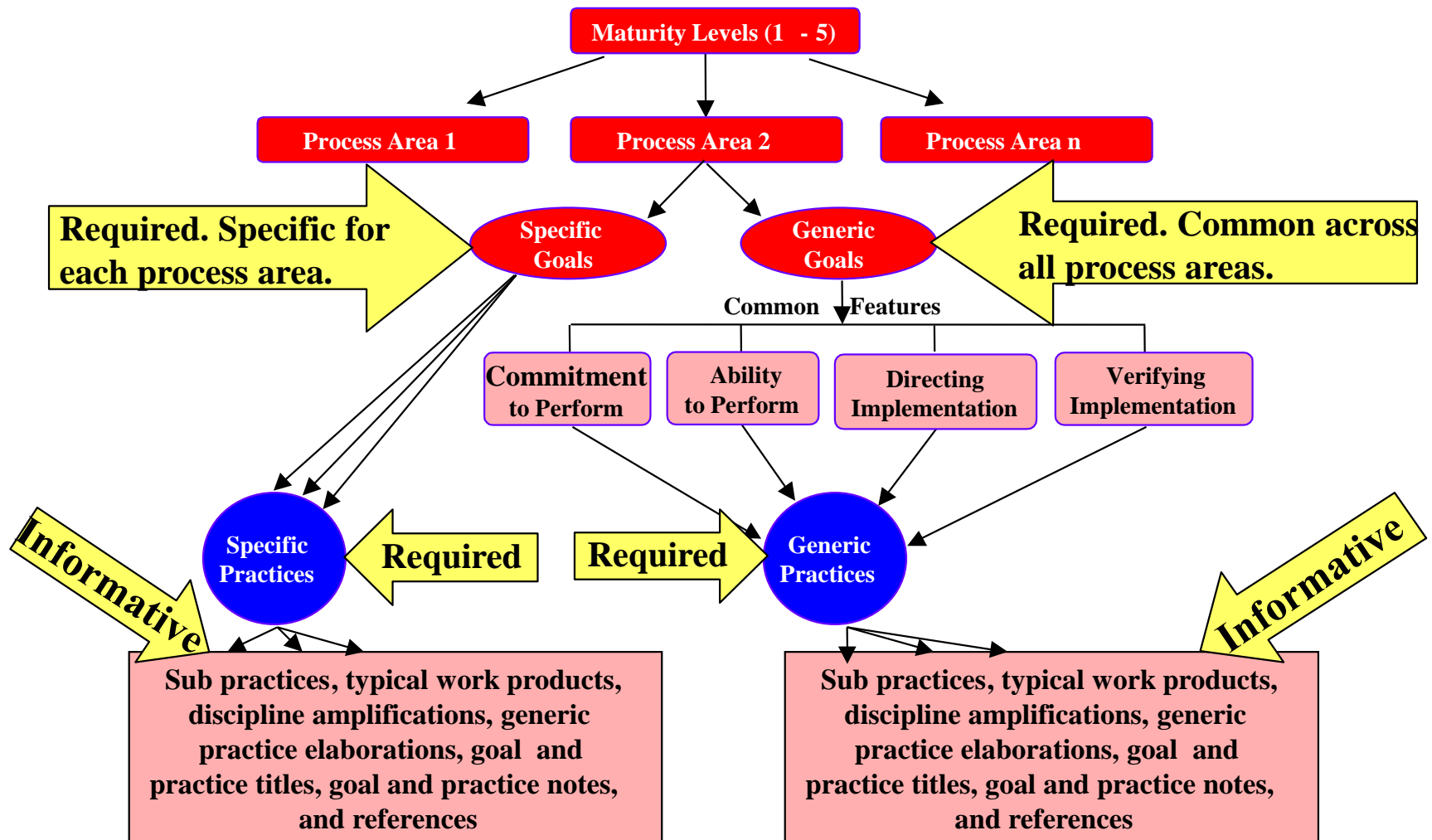
- CMMI (Capability Maturity Model Integration) is a proven industry **framework** to improve product quality and development efficiency for **both** hardware and software
 - Sponsored by US Department of Defence in cooperation with Carnegie Mellon University and the Software Engineering Institute (SEI)
 - Many companies have been involved in CMMI definition such as Motorola and Ericsson
 - CMMI has been established as a model to improve business results
- CMMI, staged, uses 5 levels to describe the maturity of the organization, same as predecessor CMM
 - Vastly improved version of the CMM
 - Emphasis on business needs, integration and institutionalization



CMMI Staged Representation - 5 Maturity Levels



CMMI Terminology & Structure



CMMI

"Staged"

Maturity Levels

Level 1 – Initial
Level 2 – Managed
Level 3 – Defined
Level 4 – Quantitatively Managed
Level 5 – Optimizing

"Continuous"

Capability Levels

Level 0 – Incomplete
Level 1 – Performed
Level 2 – Managed
Level 3 – Defined
Level 4 – Quantitatively Managed
Level 5 – Optimizing

➤ Process Management
➤ Project Management
➤ Engineering
➤ Support

Process Areas (22)

**Process Areas (22)
Active PA Subset**

Specific
Goals

Specific
Practices

**Realm
of
Appraisals**

Generic
Goals

Generic
Practices

There are no
"shalls".

Appraisals
yield
Ratings

It's a Model,
not a specification

It can be hierarchical.
It's highly detailed..

It's complex, but
nothing is complicated.

Guiding Principles of CMMI V2.0

- Focus of delivering business value
- Focus on business performance and results
- Flexible, scalable architecture
- Reduce redundancy
- Use of plain language
- Increased use of graphics and iconography for understanding and adoption



Adoption and Transition Guidance – Six Step Approach

1. Learn: Learn how CMMI will benefit the organization
2. Establish Objectives: Establish performance improvement objectives aligned to your organizational objectives
3. Analyze: Map current organizational processes to CMMI
4. Develop Action Plan: Develop and follow action plans and keep them updated
5. Deploy Improvements: Deploy improvements and measure results
6. Appraise capability: Appraise capability and assess performance



Maturity Levels Should Not Be Skipped

- Each maturity level provides a necessary foundation for effective implementation of processes at the next level.
 - Higher level processes have less chance of success without the discipline provided by lower levels.
 - The effect of innovation can be obscured in a noisy process.
- Higher maturity level processes may be performed by organizations at lower maturity levels, with the risk of not being consistently applied in a crisis.



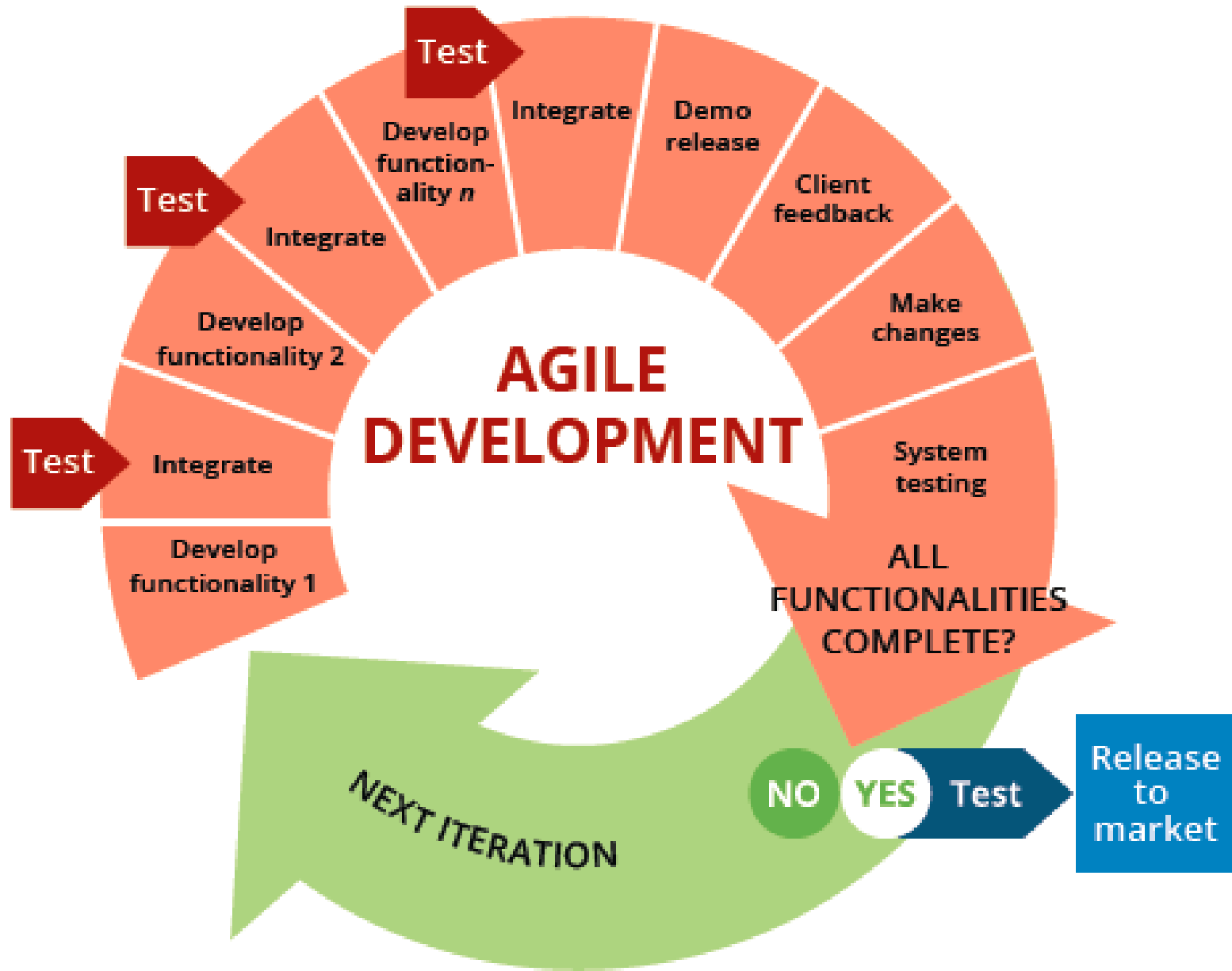
Agile Software Development

February 2001 (Agile Manifesto)

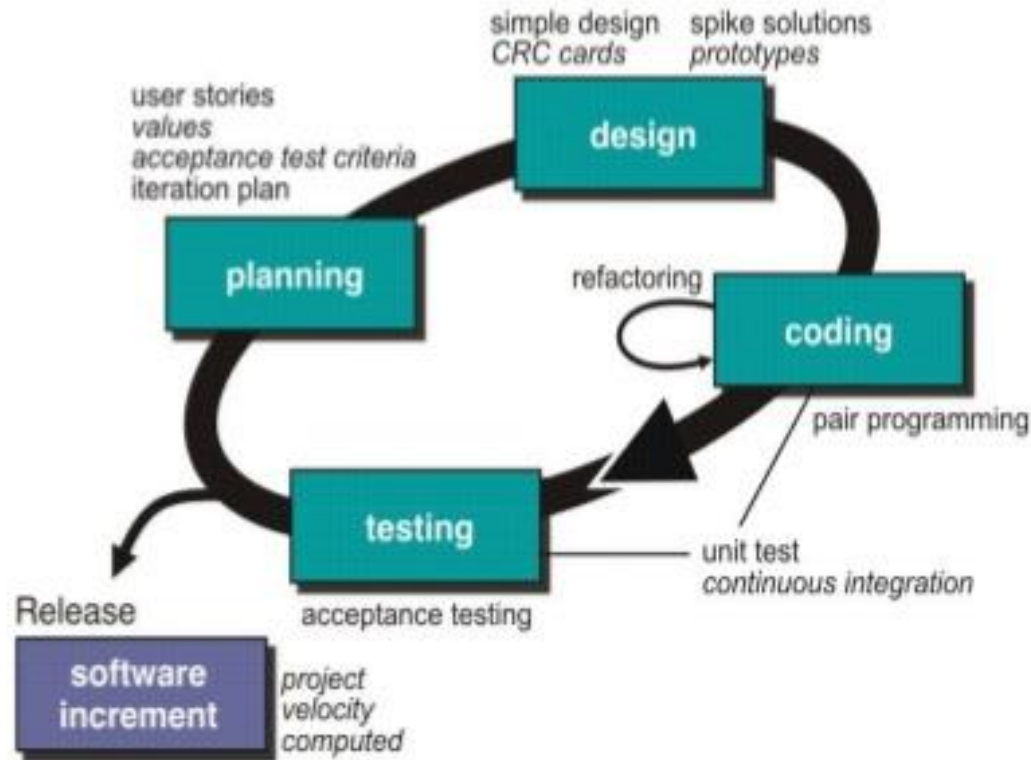
The value of ASD

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan





Extreme Programming (XP)



Framework Assumptions and Focuses

- Agile
 - How
 - Project
 - Learning in projects
 - Short-term view
- CMMI
 - What
 - Organization
 - Learning at many levels
 - Long-term view



CMMI and Agile Are Perceived to be at Odds with Each Other



Reliant on process definitions, measures, and artifacts

Predictive

Generally driven by management

Agile Alliance

Skeptical of process definitions, measures, and artifacts

Emergent

Often begun as grass-roots effort

CMMI and Agile proponents are apt to demonize or ignore each other.



What is Pragmatic Agility®?

SCRUM

(Management Framework)

- Product Backlog
- Sprint Planning
- Daily Scrum
- Burn-down Chart
- Sprint Review
- Sprint Retrospective

Benefit

- Visibility
- Predictability
- Adaptability
- Value

Extreme Programming (XP)

(Development Methodology)

- Continuous Testing
- Pair Programming
- Co-Location
- Code Refactoring
- Continuous Integration
- Collective Ownership

Benefit

- Quality
- Adaptability
- Efficiency

CMMI

(Continuous Process Improvement)

- Defined Processes
- Defined Metrics
- Process Measurement
- Process Improvement

Benefit

- Quality
- Predictability



Remember

- The CMMI model is not a process.
- This model shows what to do,
- Not how to do it or who does it.

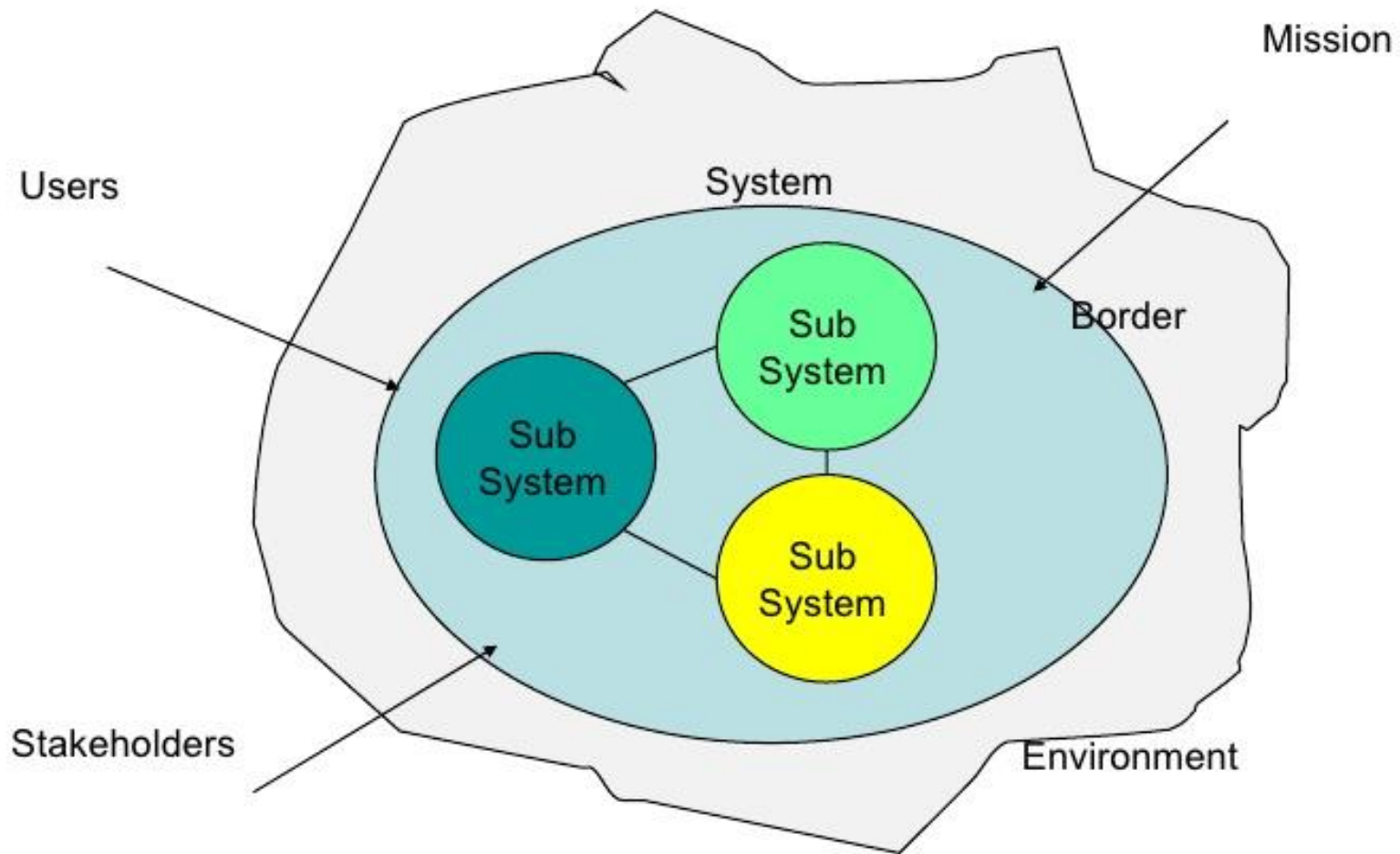




Contents

- Software Architecture
- CMMI and Agile Software Development
- The Systems View of Project Management
- Understanding Organizations
- Project Phases and Project Life Cycle
- The Context of IT Projects
- The Recent Trends of IT Projects

Systems, Systems Philosophy, Systems Analysis, Systems Management and Systems Approach



Systems Management Approach

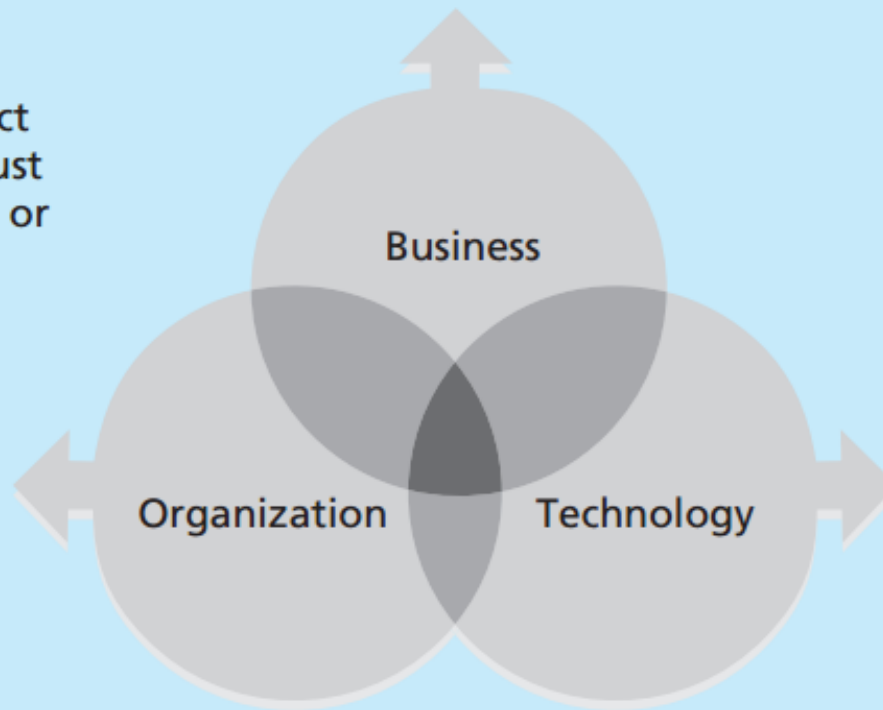
- Projects cannot be developed in isolation, projects must operate in a broad organizational environment.
- Systems management addresses three issues:
 - Business issues
 - Technological issues
 - Organizational issues





清华大学
Tsinghua University

- What will the tablet project cost the college?
- What will it cost students?
- What will support costs be?
- What will the impact be on enrollments?



- Will the tablet project affect all students, just traditional students, or only certain majors?
- How will the project affect students who already have tablets or laptops?
- Who will develop special applications or books for the tablets?
- Who will train students, faculty, and staff?

- Should the tablets be based on Apple, Microsoft, Android, or another system?
- What applications will be required?
- What will the hardware specifications be?
- How will the tablets affect various networks and speed?
- Will more power cords be required in the classroom?





Contents

- Software Architecture
- CMMI and Agile Software Development
- The Systems View of Project Management
- Understanding Organizations
- Project Phases and Project Life Cycle
- The Context of IT Projects
- The Recent Trends of IT Projects

How many types of organizations?

What is an Organization?

There are many types of organizations



Non-Governmental Organizations
International Organizations
Not-for-Profit Organizations
Secret Societies
Schools
Governments
Charities
Corporations
Resistance Movements
Criminal Organizations
Armed Forces
Voluntary Associations
Political Organizations

An organized group of people who have a common goal and work together

Four Frames of Organizations

Structural frame: Roles and responsibilities, coordination, and control. Organizational charts help describe this framework

Human resources frame: Providing harmony between needs of the organization and needs of people

Political frame: Coalitions composed of varied individuals and interest groups. Conflict and power are key issues.

Symbolic frame: Symbols and meanings related to events. Culture, language, traditions, and image.

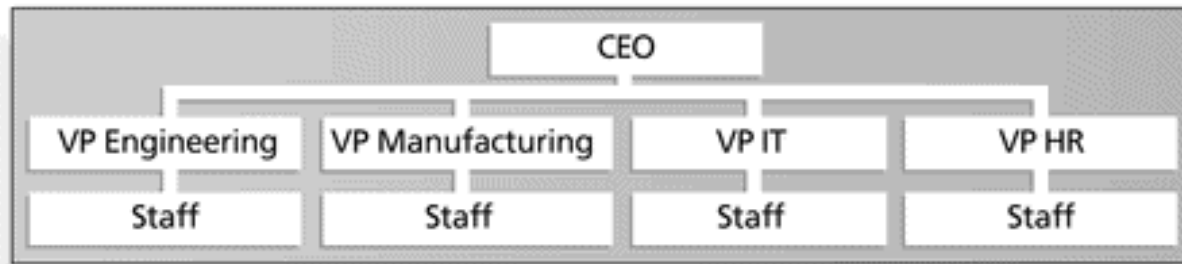


Organizational Structures

- Functional: functional manager or vice presidents reporting to CEO
- Project: program managers report to the CEO
- Matrix: people often report to both a functional manager and one or more project managers.



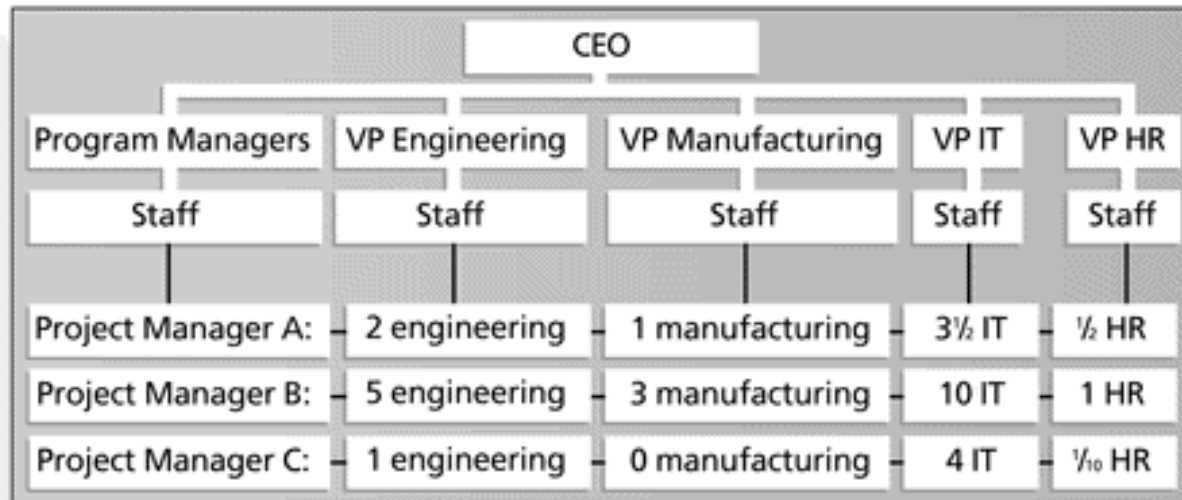
Functional



Project



Matrix

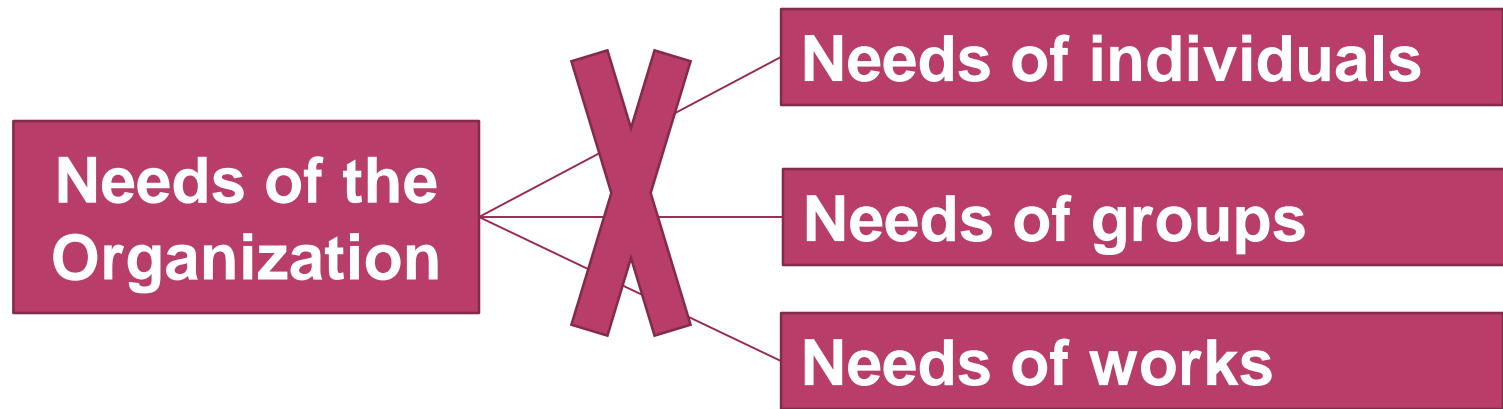


Project Characteristics						
Organiza- tional Struc- ture Type	Work Groups Arranged by:	Project Manager's Authority	Project Manager's Role	Resource Avail- ability	Who Man- ages the Project Budget?	Project Manage- ment Adminis- trative Staff
<i>Organic or Simple</i>	Flexible; people working side-by-side	Little or none	Part-time; may or may not be a des- ignated job role like coordinator	Little or none	Owner or operator	Little or none
<i>Functional (centralized)</i>	Job being done (e.g., engineering, manufacturing)	Little or none	Part-time; may or may not be a des- ignated job role like coordinator	Little or none	Func- tional manager	Part-time
<i>Multi-divi- sional (may replicate functions for each divi- sion with little cen- tralization)</i>	One of: prod- uct; produc- tion processes; portfolio; program; geo- graphic region; customer type	Little or none	Part-time; may or may not be a des- ignated job role like coordinator	Little or none	Func- tional manager	Part-time
<i>Matrix - strong</i>	By job func- tion, with proj- ect manager as a function	Moderate to high	Full-time designated job role	Moder- ate to high	Project manager	Full-time

<i>Matrix - weak</i>	Job function	Low	Part-time; done as part of another job and not a designated job role like coordinator	Low	Functional manager	Part-time
<i>Matrix - balanced</i>	Job function	Low to moderate	Part-time; embedded in the functions as a skill and may not be a designated job role like coordinator	Low to moderate	Mixed	Part-time
<i>Project-oriented (composite, hybrid)</i>	Project	High to almost total	Full-time designated job role	High to almost total	Project manager	Full-time
<i>Virtual</i>	Network structure with nodes at points of contact with other people	Low to moderate	Full-time or part-time	Low to moderate	Mixed	Could be full-time or part-time
<i>Hybrid</i>	Mix of other types	Mixed	Mixed	Mixed	Mixed	Mixed
<i>PMO®</i>	Mix of other types	High to almost total	Full-time designated job role	High to almost total	Project manager	Full-time



Human Resource Frame



Two important issues:

- Shortage of skilled information technology workers
- Unrealistic schedules imposed on many projects



Politic Frame

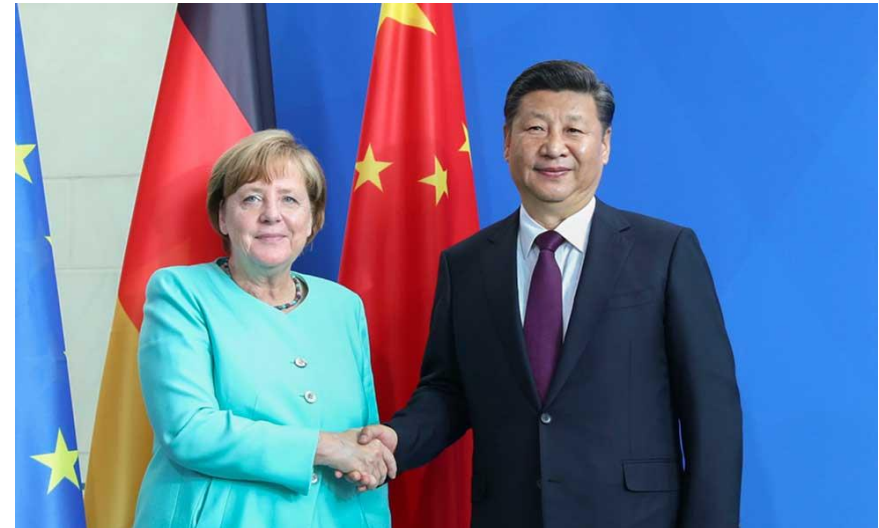


- Scarce Resources
- Leadership
- Power



Symbolic Frame

- Was it a good sign that the CEO came to a kickoff meeting for a project, or was it a threat?
- How do people dress?
- How many hours do they work?
- How do they run meetings?



Organizational Culture

A set of shared assumptions, values and behaviors that characterize the functioning of an organization.

Same organization can have different sub-cultures



10 Characteristics of Organizational Culture

- Member identity
- Group emphasis
- People focus (balance)
- Unit integration
- Control (balance)
- Risk Tolerance
- Reward criteria
- Conflict tolerance
- Means-ends orientation (balance)
- Open-systems focus

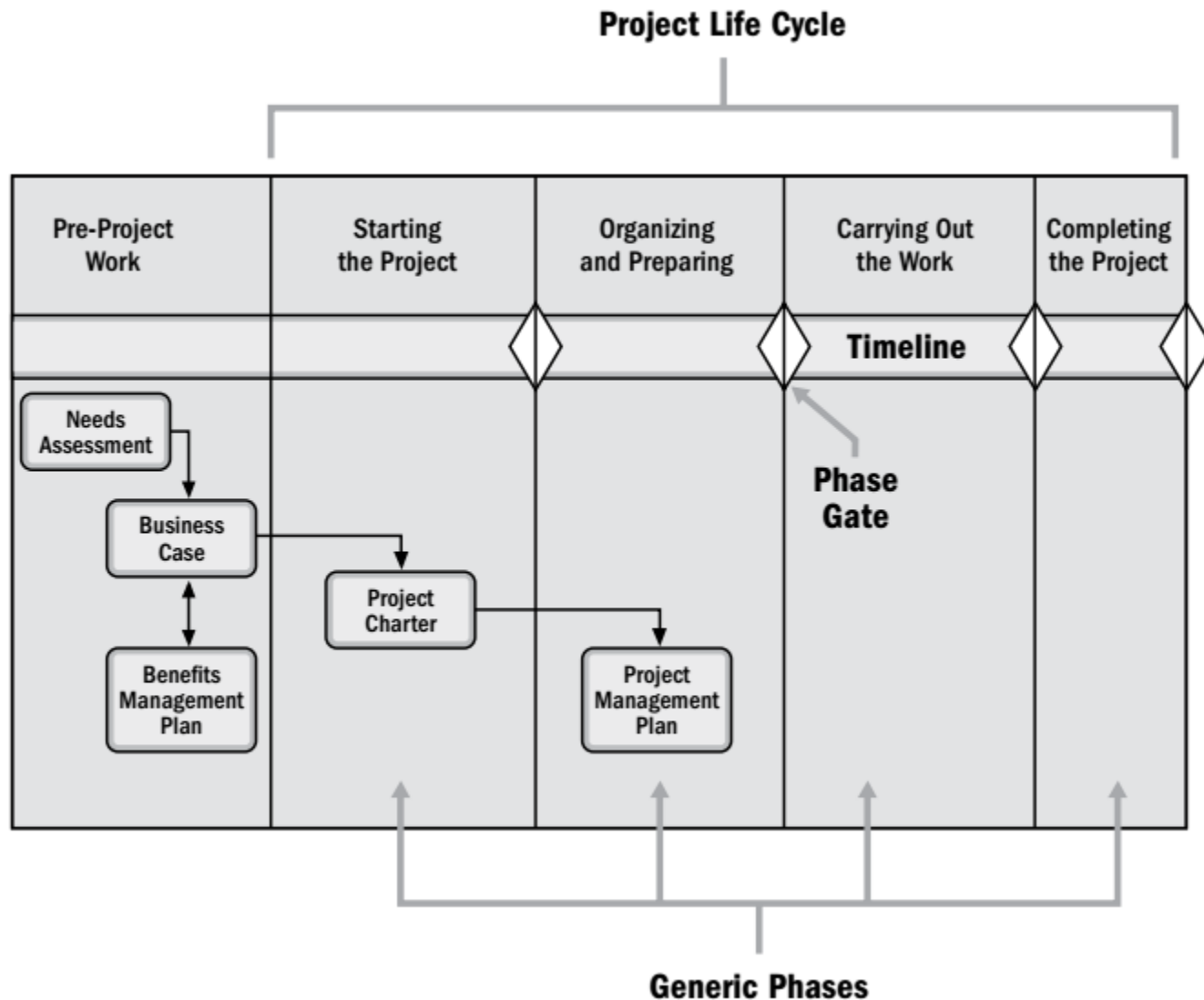




Contents

- Software Architecture
- CMMI and Agile Software Development
- The Systems View of Project Management
- Understanding Organizations
- Project Phases and Project Life Cycle
- The Context of IT Projects
- The Recent Trends of IT Projects

Project Phases vs Project Life Cycle



Why Focus on Process?

- Process provides a constructive, high-leverage focus...
 - as opposed to a focus on people
 - Your work force, on the average, is as “good” as it is trained to be.
 - Working harder is not the answer.
 - Working smarter, through process, is the answer.
 - as opposed to a focus on technology
 - Technology applied without a suitable roadmap will not result in significant payoff.
 - Technology provides the most benefit in the context of an appropriate process roadmap.



Importance of Phases

Management reviews keep projects on track and provide phase exits or kill points that enable a project to be:

- Continued
- Redirected
- Terminated

Depending on the projects effect on the organisation or possible changes in the organisation that may effect the project.



Product Life Cycle

A product life cycle is the series of phases that represent the evolution of a product, from concept through delivery, growth, maturity, and to retirement.

- Predictive life cycle
- Iterative life cycle
- Incremental life cycle
- Adaptive life cycle
- Hybrid life cycle



Software

- Programs
- Documentation during the development of programs (e.g. specification)
- Primary aids for running the programs (e.g. user manuals)
- Secondary aids for running the programs (e.g. key boards overlays)



What do we intend?

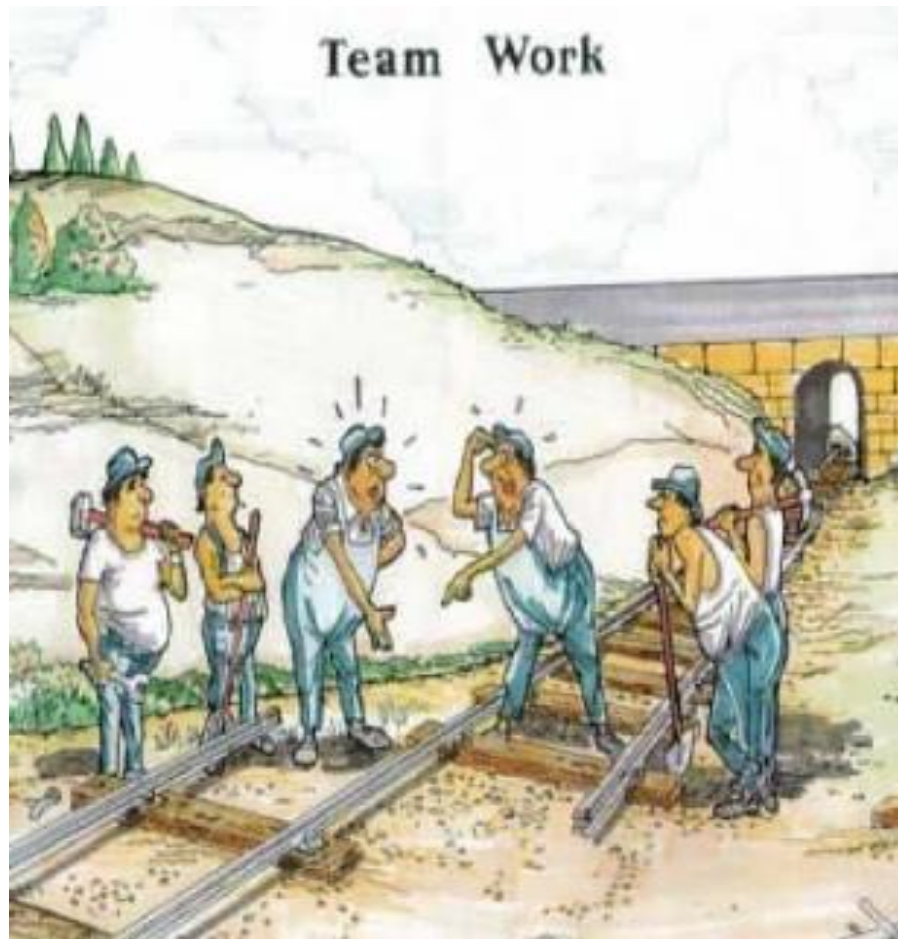
Requirements



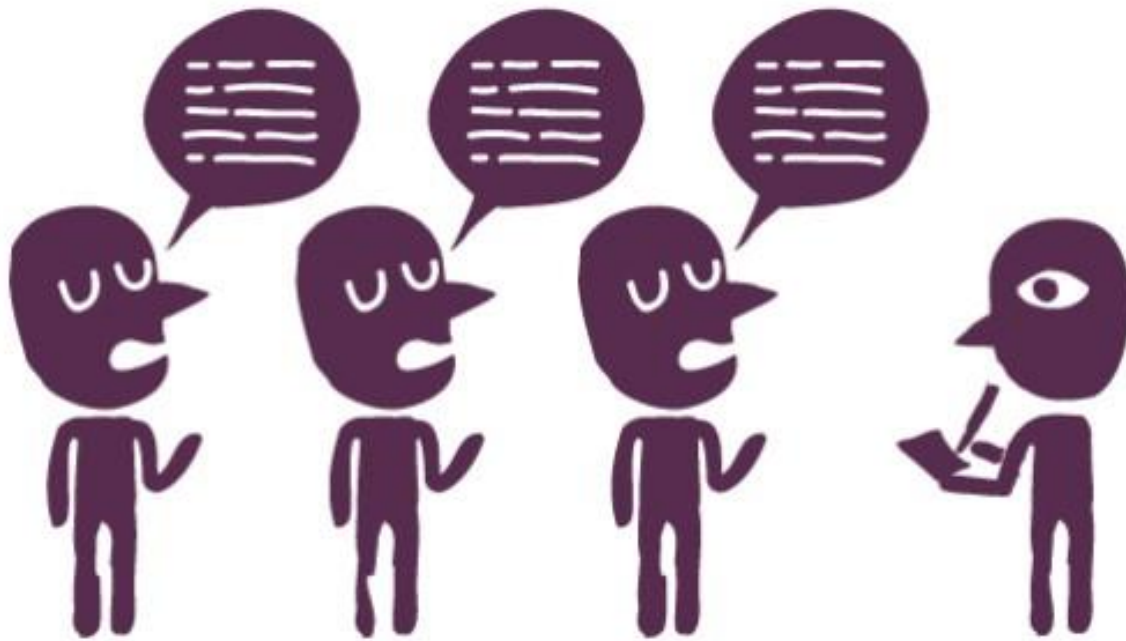
Software



How well is the industry doing ?



How to Collect Requirements?



How to Collect Requirements?



Simple Business Requirements Document Template

PROJECT DETAILS

Project Name	Enter Project Name
Project Type	(e.g. New Initiative or Phase II)
Project Start Date	
Project End Date	
Project Sponsor	
Primary Driver	(e.g. Mandatory or Efficiency)
Secondary Driver	
Division	
Project Manager/Dept	

OVERVIEW

This document defines the high level requirements [insert project name]. It will be used as the basis for the following activities:

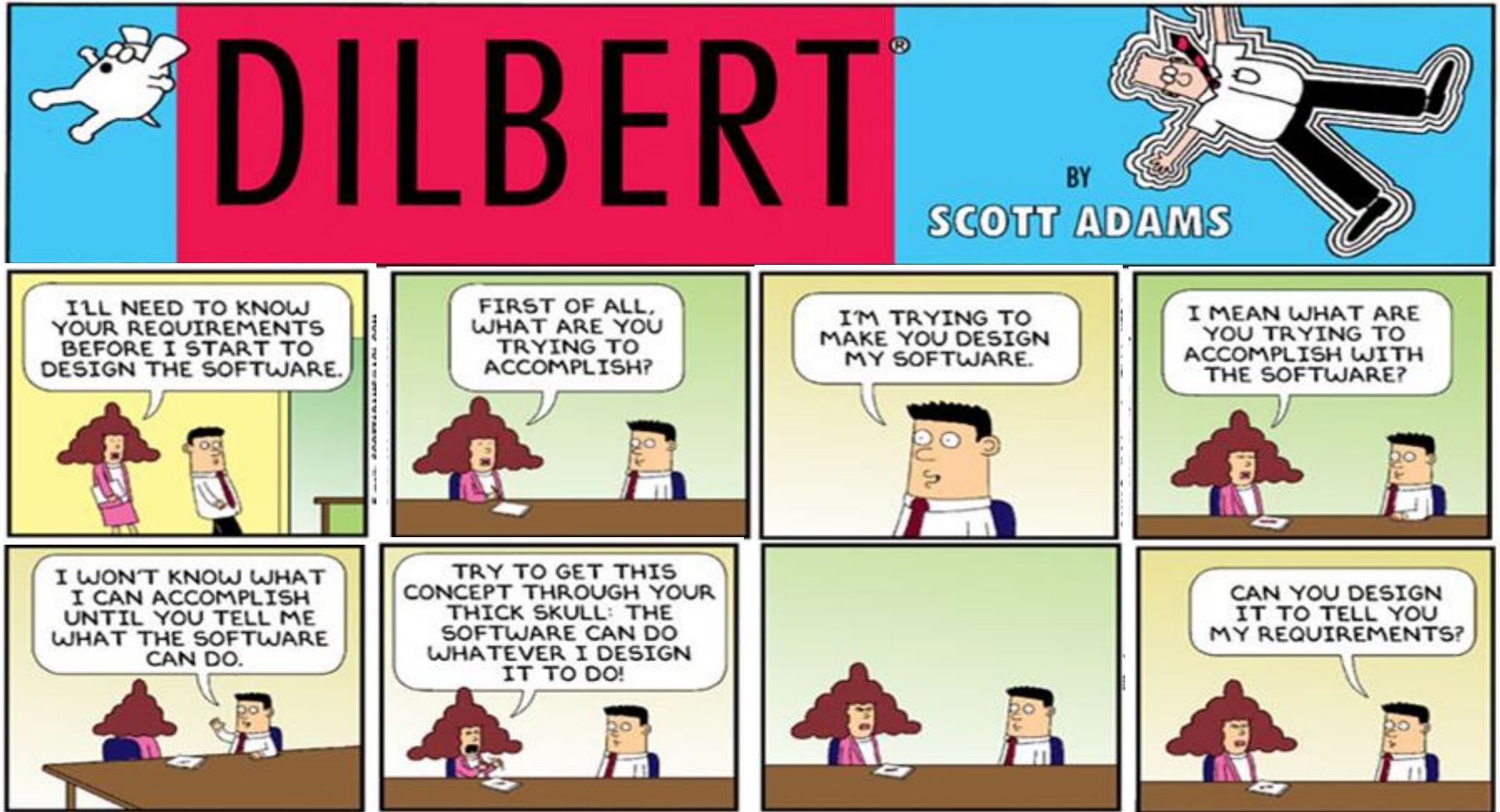
- Creating solution designs
- Developing test plans, test scripts, and test cases
- Determining project completion
- Assessing project success

DOCUMENT RESOURCES



Name	Business Unit	Role
<Identify all stakeholders and resources involved in gathering requirements>		

Requirements



In the Software Development

- Project Leader
- Analyst
- Programmer
- Business Consultant
- Document
- Installation
- Bill
- Support





How the customer explained it



How the Project Leader understood it



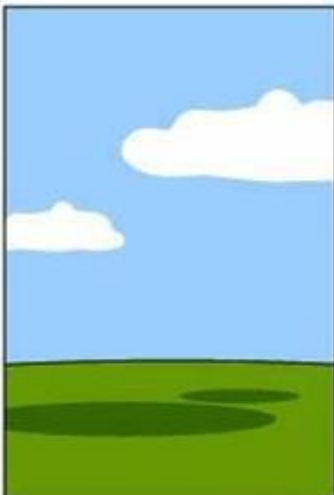
How the Analyst designed it



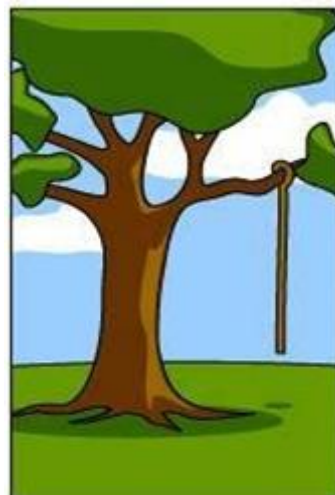
How the Programmer wrote it



How the Business Consultant described it



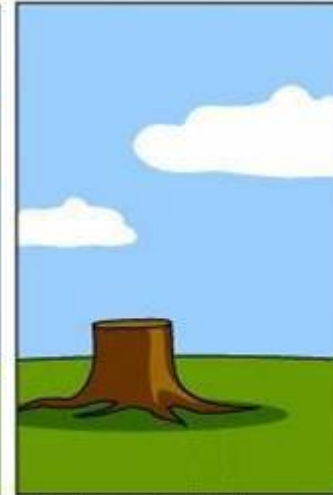
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

Building High Quality Software

- Goal is user satisfaction
 - How do we determine system is ready for delivery
 - Is it now an operational system that satisfies users' needs
 - Is it correct and operating as we thought it should ?
 - Does it pass an evaluation process ?



Verification vs Validation

- Verification
 - Am I building the product right ?
 - Begin after specification accepted
- Validation
 - Am I building the right product ?
 - Subjective - is specification appropriate ? Uncover true users' needs , therefore establish proper design ?
 - Begins as soon as project starts
- Verification & validation independent of each other
 - Even if product follows spec, it may be a wrong product if specification is wrong
eg: report missing, initial design no longer reflect current needs
 - If specification is informal, it is difficult to separate verification and validation



Inherent Problems with Software Development

- Requirements are complex
 - The client does not know the functional requirements in advance
- Requirements may be changing
 - Technology enablers introduce new possibilities to deal with nonfunctional requirements
- Frequent changes are difficult to manage
 - Identifying milestones and cost estimation is difficult
- There is more than one software system
 - New system must be backward compatible with existing system (“legacy system”)
 - Phased development: Need to distinguish between the system under development and already released systems
- Let’s view these problems as the nonfunctional requirements for a system that supports software development!
 - This leads us to software life cycle modeling



Software Life Cycle

Software is like humans.



It has a life cycle. Software in a system is conceptualized first. It becomes obsolescent at the end.

The period in between is called the software life cycle.



Software Life Cycle

- Encompasses all activities from initial analysis until end of work
- Formal process for software development
 - Describes phases of the development process
 - Gives guidelines for how to carry out the phases
- Development process
 - Analysis
 - Design
 - Implementation
 - Testing
 - Deployment



Definitions

- Software lifecycle modeling: Attempt to deal with complexity and change
- Software lifecycle:
 - Set of activities and their relationships to each other to support the development of a software system
- Software development methodology:
 - A collection of techniques for building models - applied across the software lifecycle



What is a SDLC Model?

- An SDLC model breaks down the development process into a sequence of phases or stages.
- The end of a phase is called a milestone of the project. The outcome of the phase is delivered to the customer as a milestone of the project.
- Breaking down the development process makes it easier to manage.
- Each phase can be performed in various ways.



Why Use SDLC Model?

- This is not possible for complex projects since
 - There is no control on quality assurance.
 - There is no risk analysis.
 - There is no estimation of time and cost of the project.
 - There is no proper plan.

Hence,

- The project may lead to a huge financial loss for both the customer as well as the developer.
- The project may fail.
- The project may not be delivered in time with the satisfaction of the customer.



History of SDLC Models

- Once upon a time, software development consisted of a programmer writing code to solve a problem or automate a procedure.
- Nowadays, systems are so big and complex that teams of architects, analysts, programmers, testers and users must work together to create the millions of lines of custom-written code that drive our enterprises.



Typical Software Lifecycle Questions

- Which activities should I select for the software project?
- What are the dependencies between activities?
 - Does system design depend on analysis? Does analysis depend on design?
- How should I schedule the activities?
 - Should analysis precede design?
 - Can analysis and design be done in parallel?
 - Should they be done iteratively?



SDLC MODELS

- The Waterfall model
- The Prototyping model
- The Evolutionary model
- The RAD (Rapid Application Development) model
- The Spiral model
-





Contents

- Software Architecture
- CMMI and Agile Software Development
- The Systems View of Project Management
- Understanding Organizations
- Project Phases and Project Life Cycle
- The Context of IT Projects
- The Recent Trends of IT Projects

The Nature of Information Technology Projects?



Diversity

- Few developers to hundreds of developers
- The requirements are diverse (hardware-oriented, software-oriented)
- Support every possible industry and business function

For PM, every project is new?



Diverse of Team Members

- Different background
 - Different skills
 - Move around different job functions
 - Not stable “army of free agents”
- ? Many companies purposely hire graduates with degrees in other fields**



Job Titles by Position:

CEO
CBO
CFO
COO
Chief Commercial Officer
Chief Digital Officer
Chief Marketing Officer
Chief Medical Officer
Chief Product Officer
Chief Scientific Officer
Chief Security Officer
Chief Technology Officer

General Counsel
VP Business Development
VP Client Services
VP Clinical Development
VP Drug Discovery &
Development
VP Engineering
VP Client Services
VP Finance
VP Human Resources
VP Intellectual Property
VP IoT
VP Manufacturing
VP Marketing
VP Medical Affairs

VP Operations
VP Product
VP Quality
VP R&D VP Regulatory
VP Sales
VP Sales - Regional
VP Sales - Global/International
VP Sales - National
VP Technology

Controller
Director Business Development
Director Clinical Trials
Director Engineering
Director Finance
Director Human Resources
Director Manufacturing
Director Marketing
Director of Quality Control
Director of Software/IT
Director Operations
Director Product
Director Regulatory
Director R&D
Director Sales
Director Sales - National
Director Sales - Regional

Director Technology
Director UX

Accountant - Sr
Accountant - Staff
Assay Development Manager
Assembler
Assistant - Administrative
Assistant - Executive
Biostatistician
Business Development
Manager
Buyer/Planner
Chemist
Chemist – Quality Control
Clinical Field Specialist
Clinical Project Manager
Clinical Research Associate
Clinical Research Coordinator
Clinical Research Manager
CNC Machinist
Compliance Officer
Computer Operators/
Data Entry
Customer Service Manager
Customer Service
Representative



Diverse Technologies

- Job titles reflect the different technologies required to hold those positions
 - Hardware vs database
 - Security vs business
 - Even for programmers
- The technologies change rapidly
 - Might close a project
 - Shorten the time frame
 - This fast-paced environment requires equally fast-paced processes to manage and produce projects and products





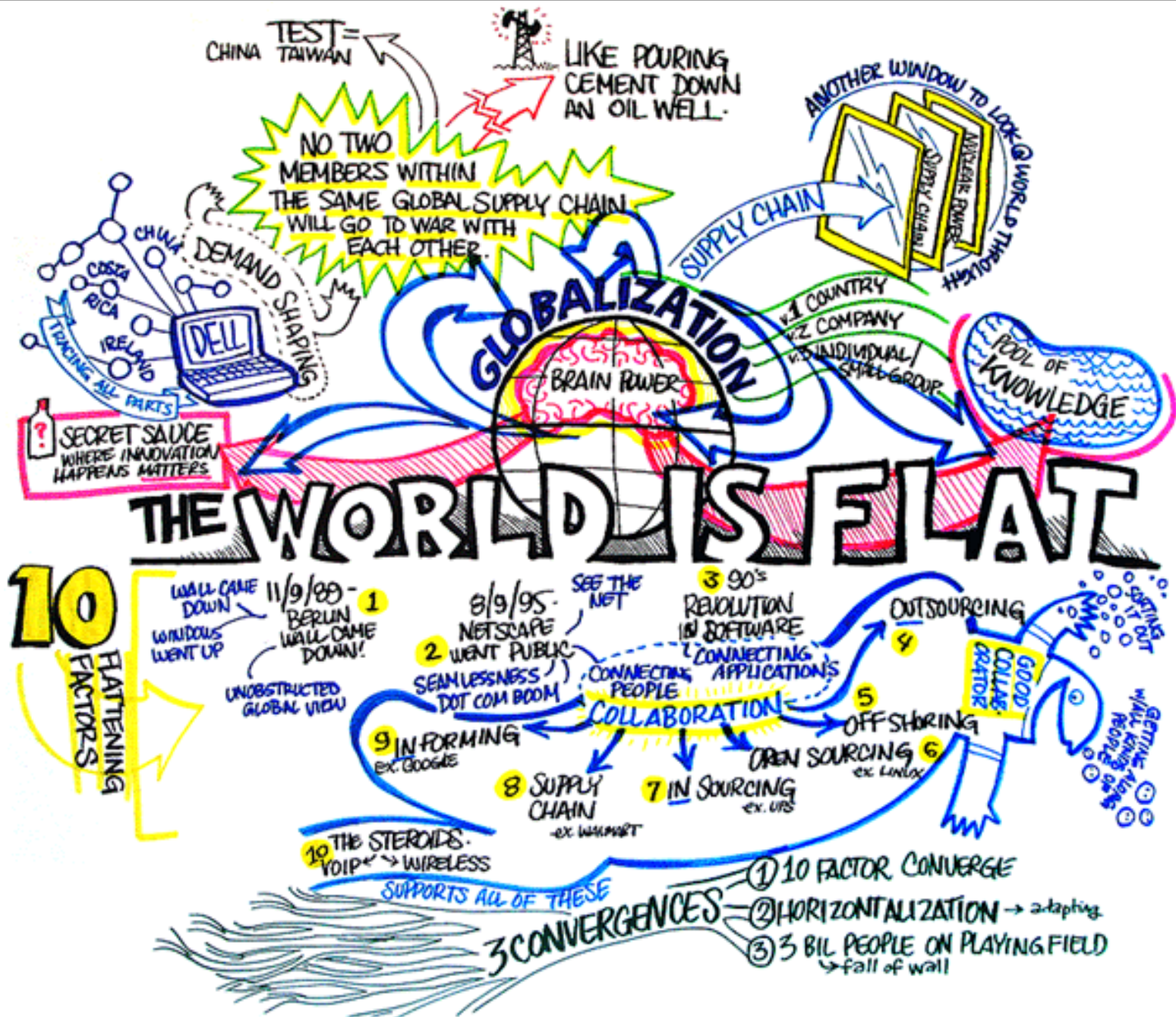
Contents

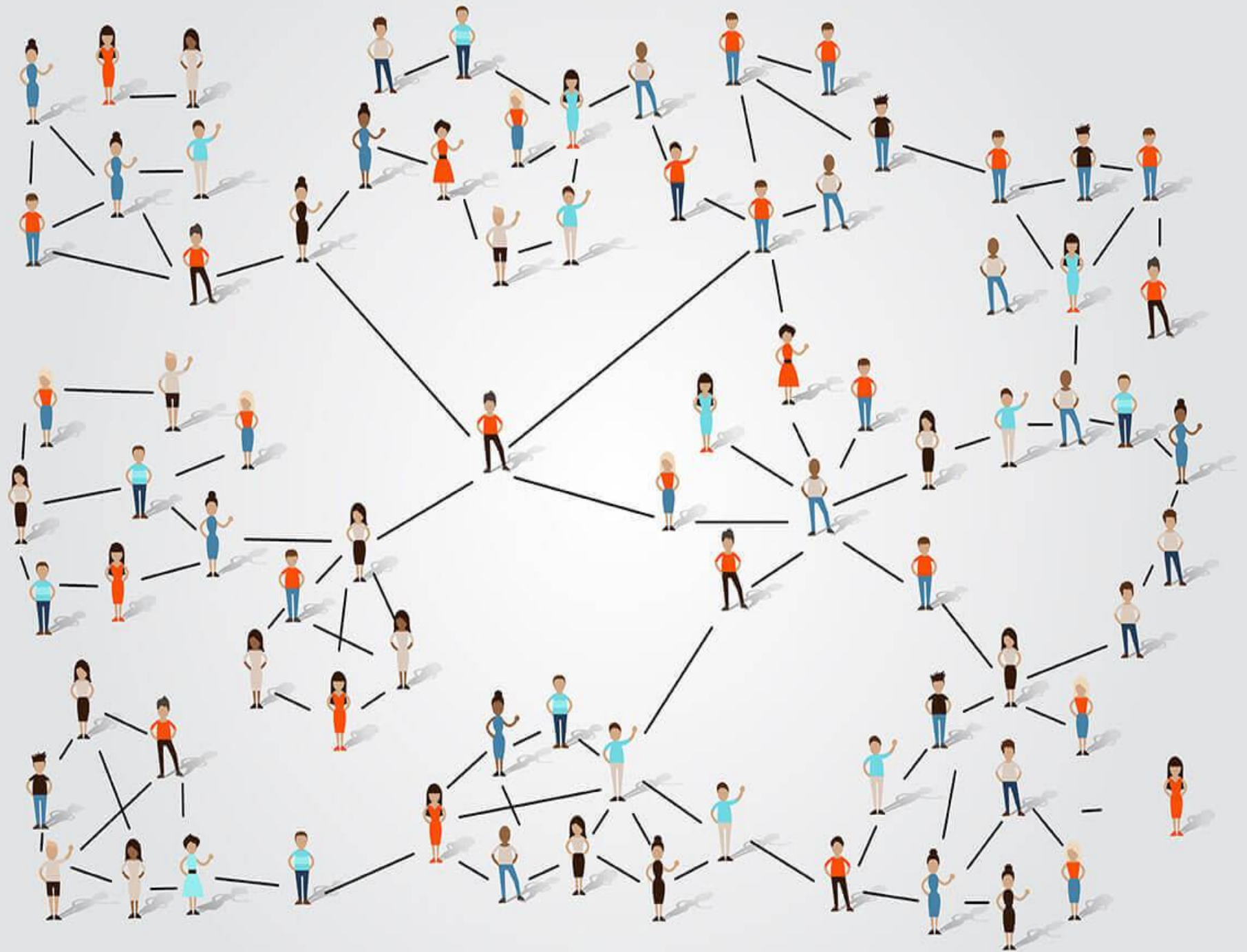
- Software Architecture
- CMMI and Agile Software Development
- The Systems View of Project Management
- Understanding Organizations
- Project Phases and Project Life Cycle
- The Context of IT Projects
- The Recent Trends of IT Projects

Recent Trends

- Globalization (“the world is flat”, Tom Friedman)
- Outsourcing
- Virtual Teams
- Agile project management







Globalization

- Communication
(time zone, language, culture, holiday...)
- Trust
- Common work practice
- Tools (enhance communications and work practices)
 - ZhuMu, Tencent Meeting, WeChat, Dingding, Skype, Google Docs, or Social Media
 - Jazz FlowDock



Outsourcing

- Offshoring
- Remain competitive
- Reduce costs



12-hour shift, 96 hours – 10K iphones

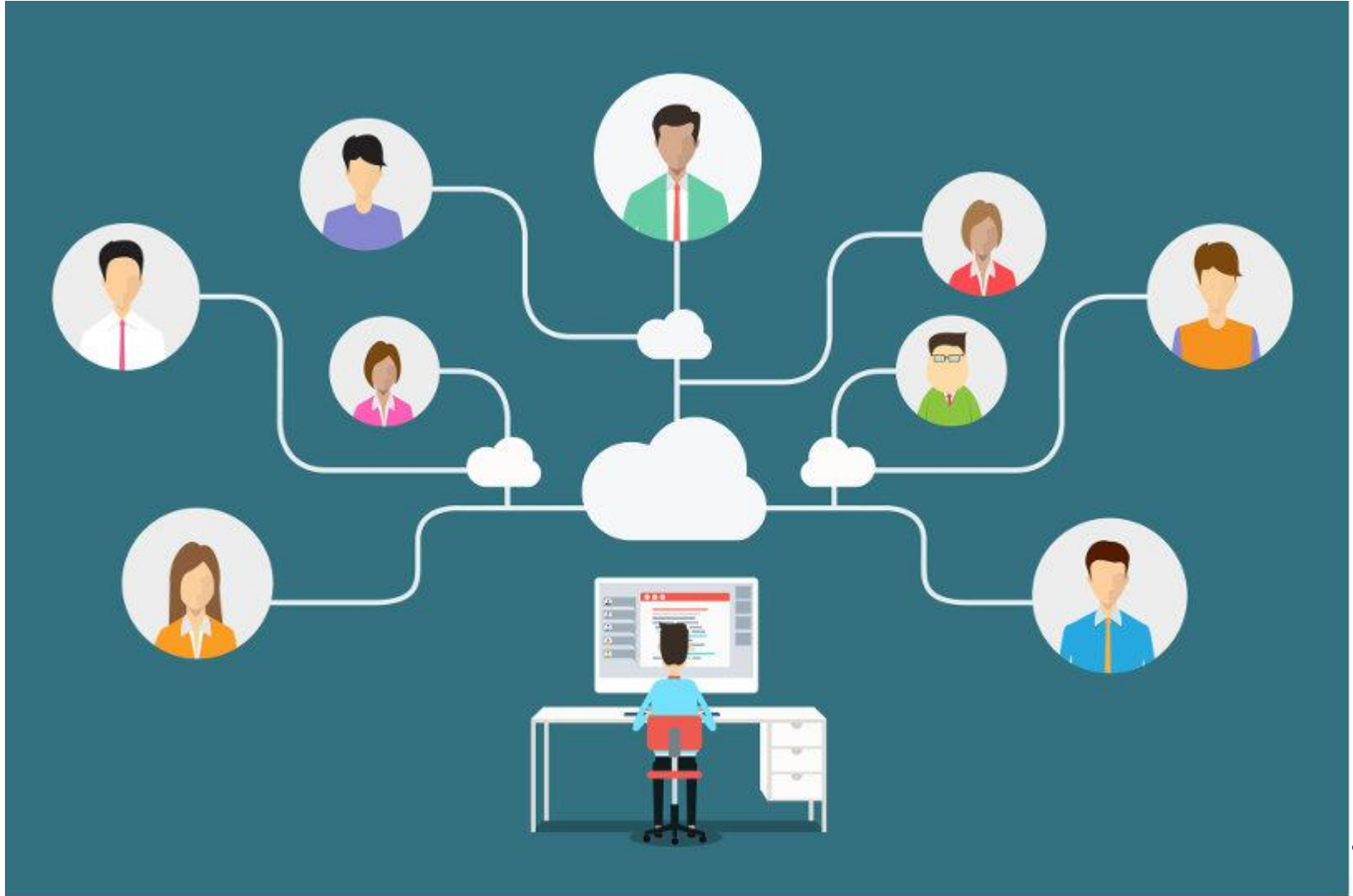


Crowd Sourcing





清华大学
Tsinghua University



Virtual Teams - Advantages

A group of individuals who work across time and space using communication technologies.

- Lowering costs
- Providing more expertise and flexibility
- Increasing the work/life balance for team members



Virtual Teams - Disadvantages

- Isolating team members
- Communication problems
- Reducing informal information transfer
- Increasing the dependence on technology



Factors Influencing Success

- Team processes
- Leadership style
- Trust and relationships
- Team member selection and role preferences
- Task-technology fit
- Cultural differences
- Computer-mediated communication
- Team life cycles
- Incentives
- Conflict management



Suggestions for Managing Global Project Teams (KPMG)

- Employ greater project discipline for global projects
- Think globally, but act locally and integrate stakeholders at all project levels
- Consider collaboration over standardization to help balance the goals and project approach
- Keep momentum going for projects
- Consider the use of newer tools and technologies



Agile Project Management

- Agile means being able to move quickly and easily
- Use a method based on iterative and incremental development, in which requirements and solutions evolve through collaboration
- Requirements are unknown or change quickly

2011, PMI-ACP (Agile Certified Practitioner)



The Agile: Scrum Framework at a glance

Inputs from Executives,
Team, Stakeholders,
Customers, Users



清华大学
Tsinghua University



Thanks!



清华大学
Tsinghua University