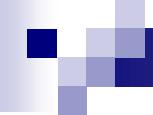


Project Management

Introduction

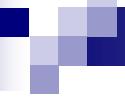
Course's Goals (theoretical)

- SE project development phases and disciplines
 - Plan Driven (e.g. RUP)
 - Agile (e.g. SCRUM)
- Acknowledge the interfaces and interdependencies between the disciplines in SE
- Present various project management techniques and their application in the two prominent methodologies
- Project Management Metrics and Indicators
- Understand the risks and the factors that lead to success or failure; Risk Management
- Reflections of Project Management on the Software Quality



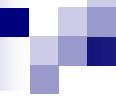
Course's Goals (practical)

- Build a project plan according to phases and disciplines
- Plan activities according to available resources
- Build a project schedule
- Prepare for and manage change
- Manage risk
- Monitor and measure project progress



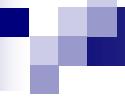
Course Structure

- Introduction to Project Management
 - Plan Driven Methodologies
 - Agile Methodologies
 - Project management techniques
- Project Manager's Role
- Planning and tailoring the project's process
- Planning the development
 - The Development Plan
 - The project structure (WBS)
 - Resources and the project schedule



Course Structure (cont.)

- Monitoring and Control
 - Metrics
 - Performance Indicators
- Risk Management
- Change Management
- People Management & What makes a good PM
- Project Closure



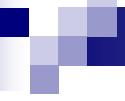
What do you expect from this course?

- [answers here]



PM Overview

- What is a Project?
- What is Project Management?
- Areas of Expertise

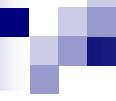


What is a project?

- A project is a **temporary** endeavor, undertaken to create a **unique** product, service, or result.

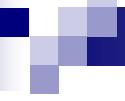
Temporary

- a definite beginning
- a definite end – When?
 - project's objectives **have been achieved**
 - it becomes clear that the project objectives **will not** or **cannot be met**
 - the **need** for the project **no longer exists** and the project is terminated
- Projects are not ongoing efforts!!



Unique Products, Services, or Results

- A product is quantifiable, and can be either an **end item** in itself or a **component** item
- A capability to perform a **service**, such as business functions supporting production or distribution
- A result, such as **findings** or **documents**..



Projects vs. Operational Work

- Common features
 - Performed by people
 - Constrained by limited resources
 - Planned, executed, and controlled
- Different features
 - operations are **ongoing** and **repetitive** vs.
 - projects are **temporary** and **unique**

Software Development Projects

- Software development life cycle
 - Full life cycle projects
 - Partial life cycle projects
- Approach driven SDP
 - Fresh development (from scratch)
 - COTS product customization/implementation
 - Porting
 - Integration
 - ...

SDP

■ Maintenance projects

- Defect repair
- Functional expansion
- Operational support
- Software modification

...

SE vs. PM

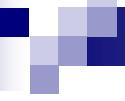
- Tightly or loosely coupled?
- SE methodologies
 - Plan-driven (UP, MSF, ...)
 - Agile (Scrum, Scaled Agile Framework® (SAFe®), eXtreme Programming (XP), Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Crystal, Lean Software Development (LSD) ...)

SE processes vs. PM processes

- Requirements Eng.
- Analysis&Design
- Construction
- Testing
- Acquisition
- Initiation
- Execution
- Closure
- +
 - Measurement & Analysis
 - Training
 - Knowledge repository

Projects and Strategic Planning

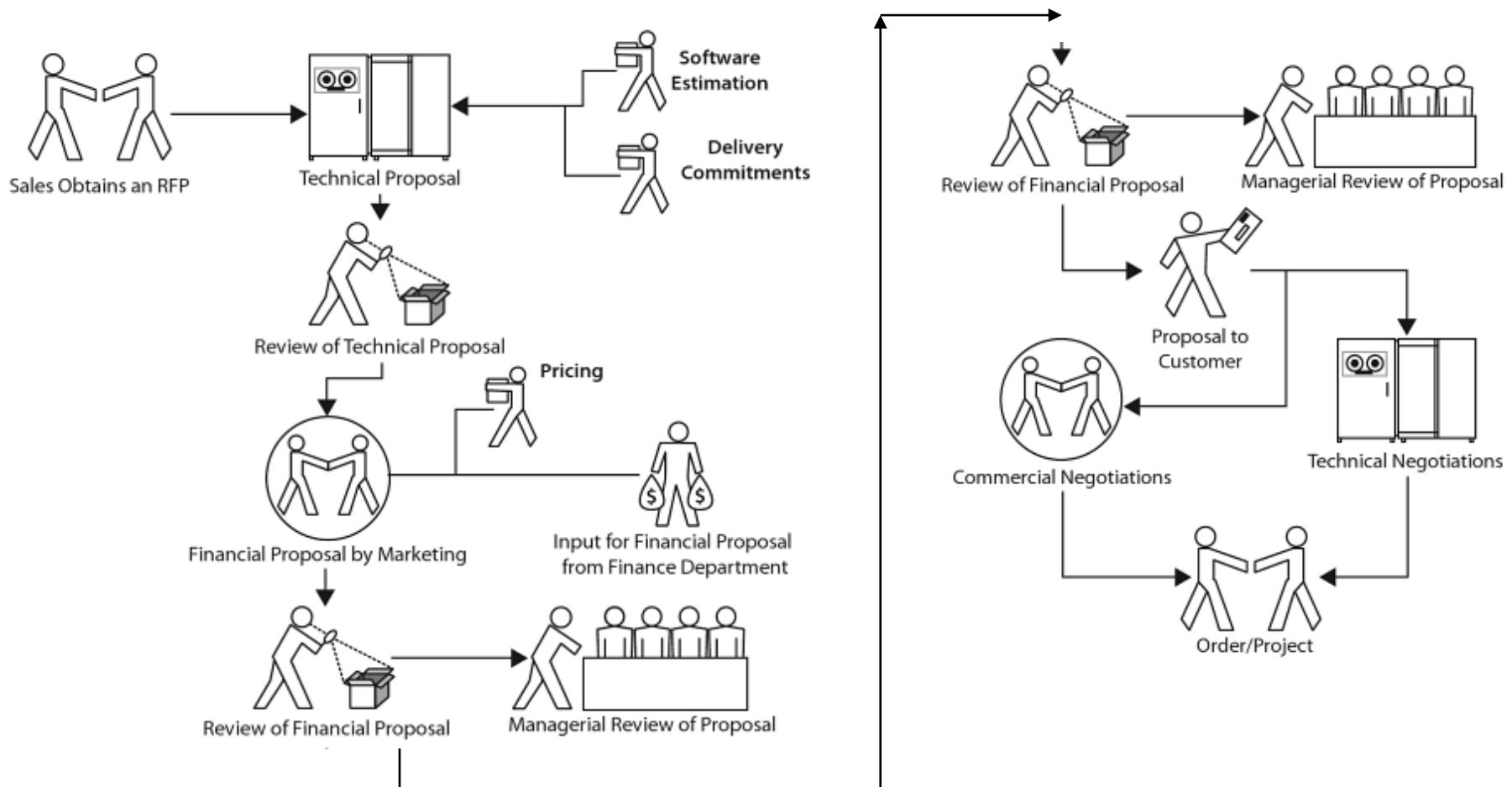
- A **market demand** (e.g., an oil company authorizes a project to build a new refinery in response to chronic gasoline shortages)
- An **organizational need** (e.g., a training company authorizes a project to create a new course in order to increase its revenues)
- A **customer request** (e.g., an electric utility authorizes a project to build a new substation to serve a new industrial park)

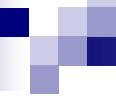


More project initiation drivers

- A **technological advance** (e.g., a software firm authorizes a new project to develop a new generation of video games after the introduction of new gameplaying devices)
- A **legal requirement** (e.g., a paint manufacturer authorizes a project to establish guidelines for the handling of a new toxic material).

Project acquisition [MSPM]



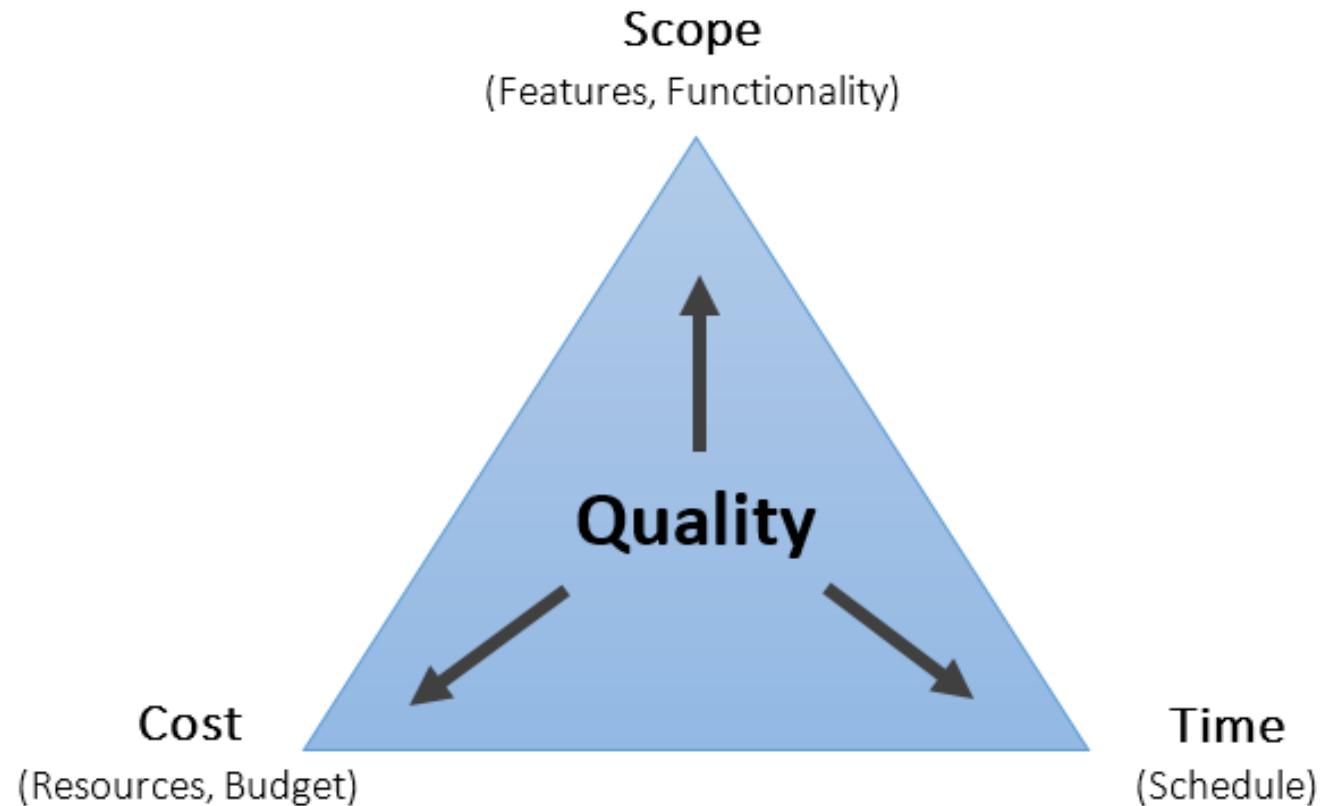


What is Project Management

- Project management is the application of
 - ❑ knowledge, skills,
 - ❑ tools and
 - ❑ techniques
- to project activities to meet **project requirements.**

Project requirements

- Scope
- Time
- Cost
- Quality
- ??



Example project – Get to the office

- 6.00 wake-up => 6.15 wake-up
- 6.10 shower => 6.30 shower
- 6.30 get dressed 50 get dressed
- 6.45 eat breakfast a muffin
- 7.10 prepare school bag for kids ey
- 7.30 in the car => 7.20 in the car
- 7.45 drop kids => 7.30 drop kids
- 8.00 in office => 7.45 in office

SUCCES???

Project budget statistics

- On average, projects go over budget by 27 % of their intended cost. ([Harvard Business Review](#))
- On average, one in six projects saw a budget overrun of 200 %. ([Harvard Business Review](#))
- 55 % of project managers cited budget overrun as a reason for project failure. ([IT-Cortex](#))
- 31 % of project managers cite meeting their budget as a criterion for project success. ([IT-Cortex](#))
- IT failure rates are estimated to be between 5–15 %, accounting for a loss of \$50–\$150 billion per year in the United States alone. ([Harvard Business Review](#))
- Projects with budgets over \$1 million have a 50 % higher failure rate than projects with budgets under \$350,000. ([Gartner](#))

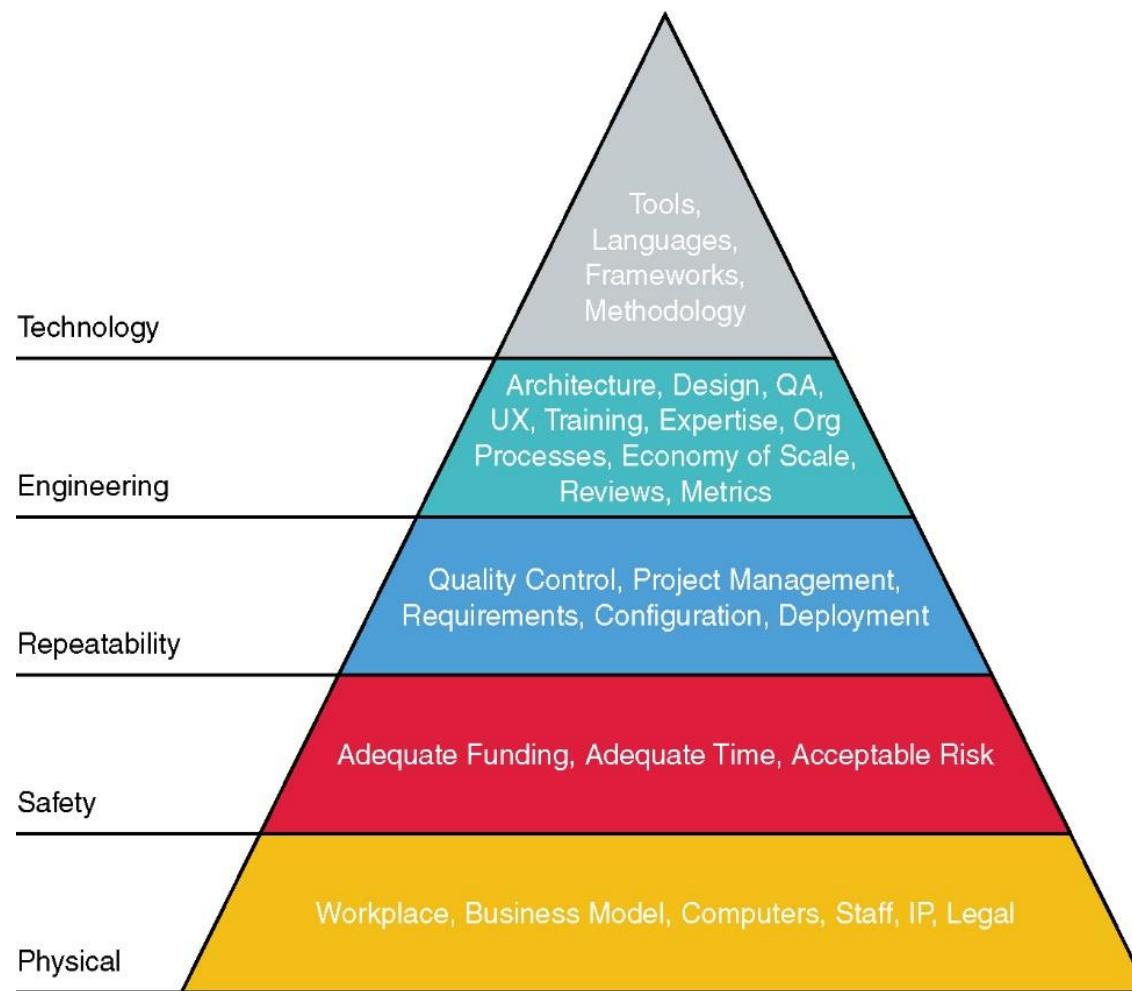
Project planning and goal statistics

- Most organizations have a 70 % project failure rate. ([4PM](#))
- Only 28 % of companies use project performance techniques. ([PMI, 2017](#))
- Only 2.5 % of companies complete 100 % of their projects successfully. ([Gallup](#))
- Only 64 % of projects meet their goals. ([Wrike](#))
- 17 % of IT projects fail so badly they can threaten the existence of a company. ([Calleam](#))
- 49 % of 840 federally funded projects were found to be poorly planned, poorly performing, or both. ([Calleam](#))
- 78 % of a group of IT and business professionals reported that their business was not aligned with project goals. ([Geneca](#))

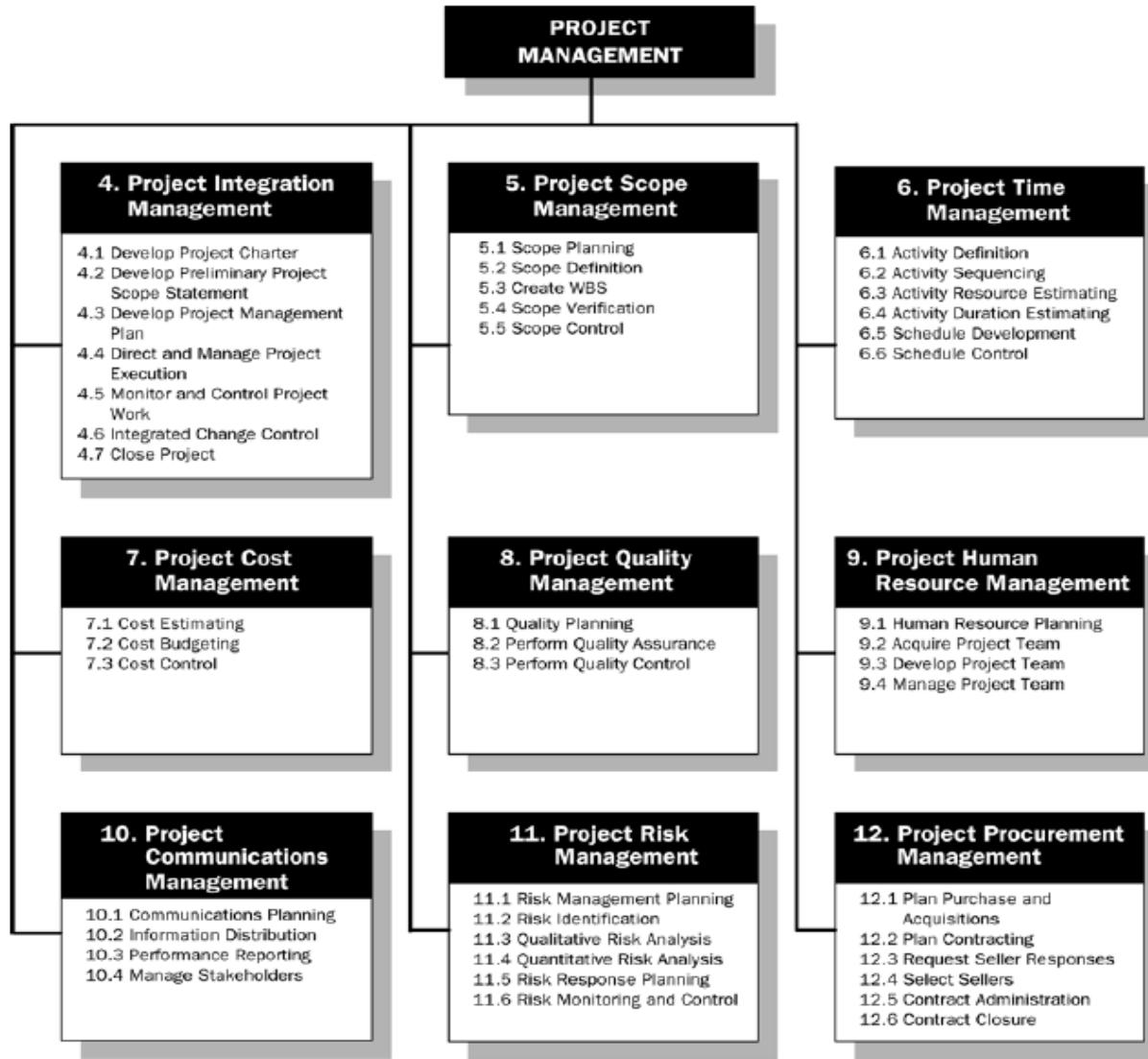
Project success statistics

- 68 percent of projects don't have an effective project sponsor to provide direction or address problems. ([Wrike](#))
- 80 percent of high-performing projects are led by a certified project manager. ([Wrike](#))
- 77 percent of high-performing projects use project management software. ([Hive](#))
- 87 percent of high-performing companies use project management software. ([Wrike](#))
- Organizations using any type of project management methodology are better at meeting budget, staying on schedule and meeting scope, quality standards and expected benefits. ([PMI, 2015](#))

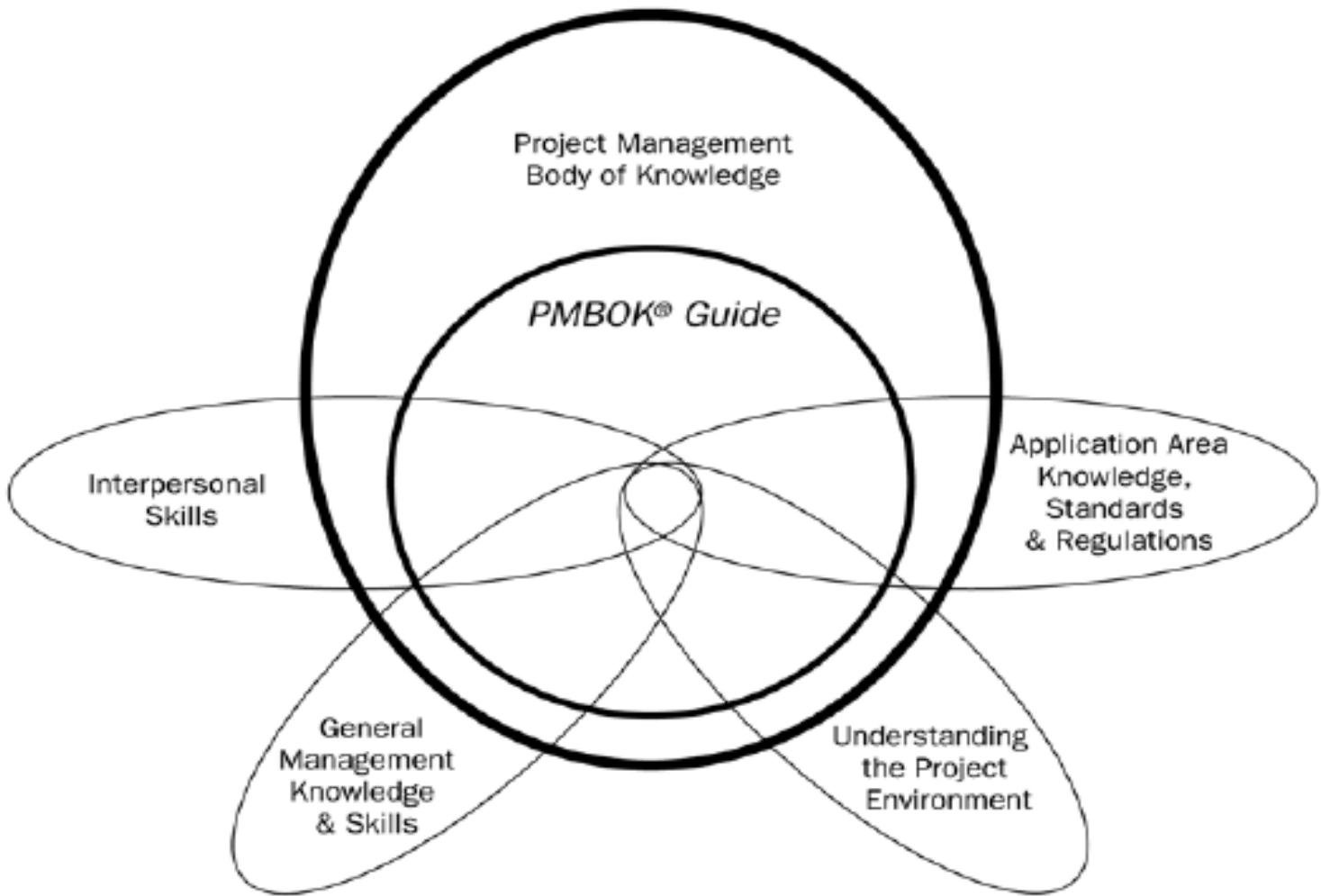
Software project hierarchy of needs



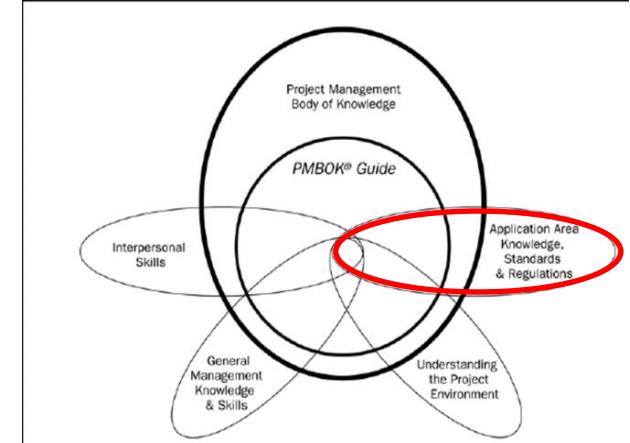
Project Management Body of Knowledge



Areas of expertise [PMBOK]



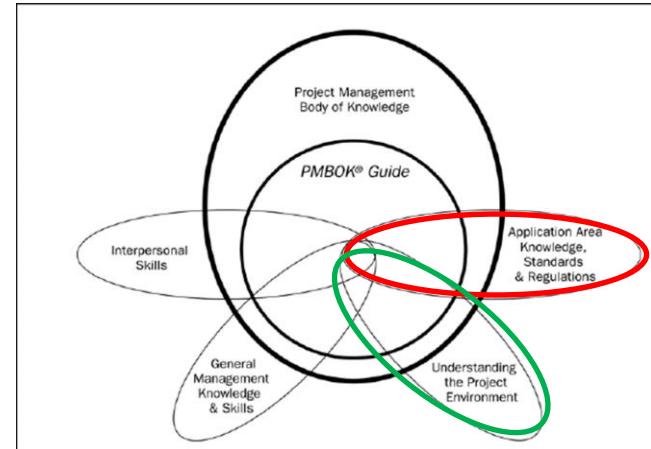
Application Area Knowledge, Standards and Regulations



- **Functional departments** and supporting disciplines, such as legal, production and inventory management, marketing, logistics, and personnel
- **Technical elements**, such as software development or engineering
- **Management specializations**, such as government contracting, community development, and new product development
- **Industry groups**, such as automotive, chemical, agriculture, and financial services.

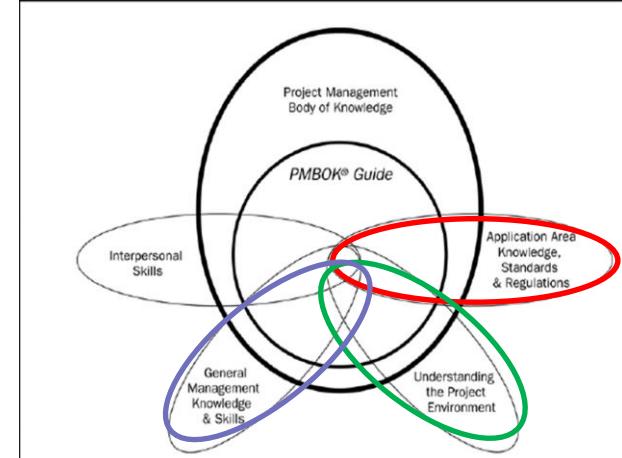
Understanding the Project Environment

- Cultural and social
- International and political
- Physical
- Etc.



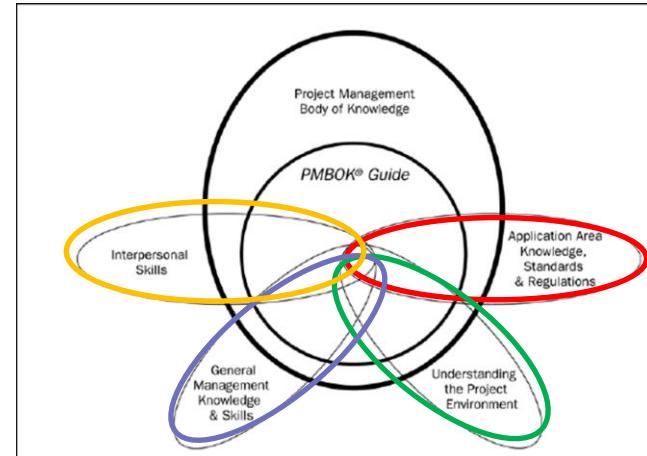
General Management Knowledge and Skills

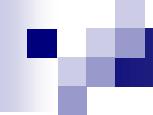
- Financial management and accounting
- Purchasing and procurement
- Sales and marketing
- Contracts and commercial law
- Manufacturing and distribution
- Logistics and supply chain
- Strategic planning, tactical planning, and operational planning
- Organizational structures, organizational behavior, personnel administration, compensation, benefits, and career paths
- Health and safety practices
- Information technology.



Interpersonal Skills

- Effective communication.
- Influencing the organization.
- Leadership.
- Motivation.
- Negotiation and conflict management.
- Problem solving.



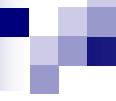


Quiz time

- Switch over to Moodle

Discussion topic next week

- Give an example of a real-life project you have already done or you plan to do. It doesn't have to be a software development project, it can be any type of project (ex. a trip, a party, etc.). Justify it is a project (1 page/2-3 slides).



Next Meeting

- PM in relation with
 - Processes
 - Software Development Methodologies

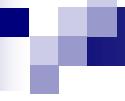


Introduction to Project Management - Part I

Lecture 2

Topics

- PM in relation with
 - Processes
 - Software Development Methodologies



Project Management Dimensions

■ Technical

- Planning
- Monitoring and Control

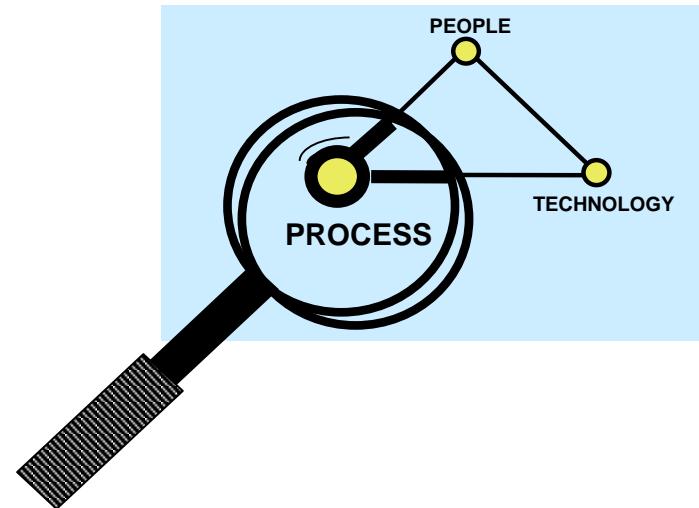
■ People

- Management
- Leadership

What is a Process? - 1

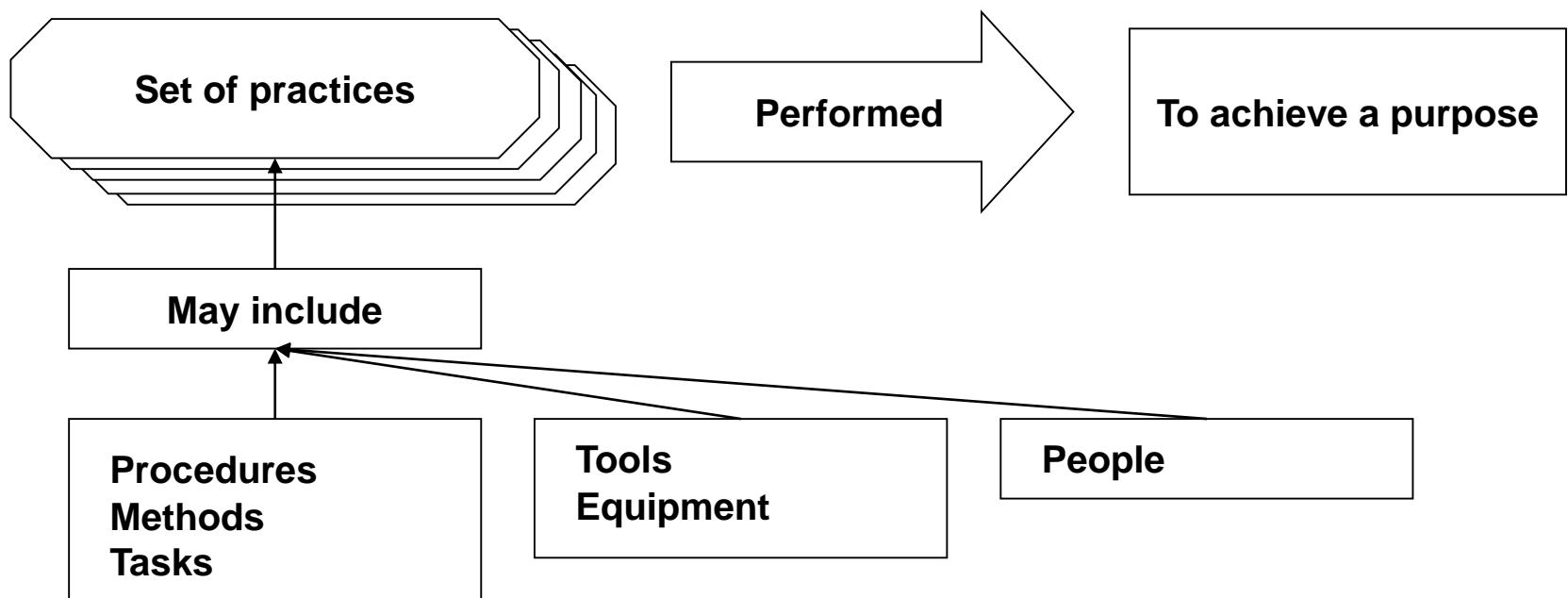
- A *process* is “*a system of operations introducing something ... a series of actions, changes, or functions that achieve an end or result*”. [Webster's dictionary]

While process is often described as a leg of the process - people - technology triad, it may also be considered the “glue” that unifies the other aspects.



What is a Process? - 2

- A *process* is a set of practices performed to achieve a given purpose; it may include tools, methods, materials, and/or people.



What is a Process? - 3

■ Characteristics of process:

- Prescribes all the major activities.
- Possesses the set of leading principles that define the objectives of the individual activities.
- The activities are organized into sequences.
- Every activity possesses the starting and ending criteria.
- Every activity uses resources in accordance with stated constraints and results with intermediate or final deliverables (products).
- Activities may be structured.



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**.

Immature Process

- Is not written down
- Is not enforced
- Is not followed
- Dependent on individuals
- Does not control requirements or products





Immature Organization



- Performance driven by the **competence** and **heroics** of the people doing the work
- **Schedule** drives everything despite requirements
- Often the schedules are **unrealistic**
- Testing **unpredictable** and often shortened
- Development starts before requirements are stable
- **Unpredictable** performance
- Few to **no** objective **measures**

Mature Process



- Is defined, documented, understood, used, and improving
- Is controlled and compliance is monitored
- Collects and uses measurement



Mature Organization

- **Requirements** are documented and **managed**
- Schedules **negotiated** on the basis of **agreed to requirements**
- Less dependent on heroics
- **Measurement** drives decisions
- **Risks** are identified and **managed**
- In times of crisis, **relies more on process**
- **Predictable** performance
- Early **defect identification** and removal

Concepts of Process Maturity



Software Process: a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products.



Underlying Premise: “*The quality of a product is largely determined by the quality of the process that is used to develop and maintain it*“.

Based on TQM principles as taught by Shewhart, Juran, Deming and Humphrey.

Levels of process

- Measure the process level against a set of best practices resulted from industry experience
- The recognized metric is CMMI (Capability Maturity Model Integration)
 - Developed at SEI Carnegie Mellon
 - Official site: <http://cmmiinstitute.com>

CMMI Overview

■ What is CMMI?

A set of practices covering:

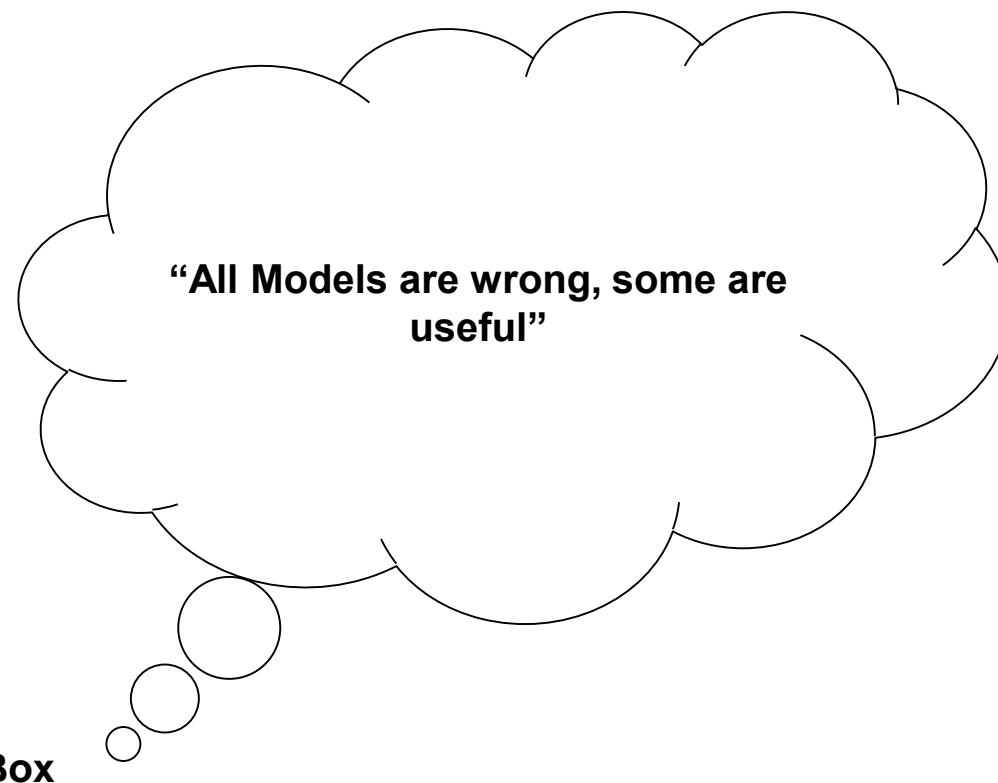
- Development
- Services
- Supplier Management
- People Management
- Data Management.

What is CMMI?

- Describes **What** not **How**
- Is an organized structured set of requirements for implementing **industry proven practices**
- Aids **progress measurement** by providing a “target”
- Provides a way to **benchmark** the maturity of your process and compare to other organizations
- Establishes a **common vision** and language for an organization

What is CMMI?

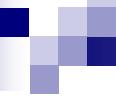
- It is a model



WE'RE GOING TO USE
CMMI. IT'S A MODEL
FOR DEVELOPING A
PROCESS TO CREATE
A FRAMEWORK.

OR IT MIGHT BE A
PROCESS FOR CREATING
A FRAMEWORK TO
MAKE A MODEL.

THERE'S NO BUDGET
FOR TRAINING, SO
WE'LL BE RELYING
ON GUESSING MORE
THAN USUAL.



What CMMI is NOT?

- A recipe book
- A complete set of practices
- A set of processes/procedures
- A set of “bolt-on processes” to be used only in times of crisis. CMMI provides a consistent, enduring framework that accommodates new initiatives.
- Restricted to a single discipline (I.e. Software Engineering or Systems Engineering). CMMI focuses on the total software intensive system problem and integrates multiple disciplines into one process-improvement framework that eliminates inconsistencies and reduces duplication

Why CMMI?

- CMMI provides guidance for **improving organization's processes** and your ability to manage the development, acquisition, and maintenance of *products or services*
- Consistently provide the right software and system, on time and on cost
- Gain visibility into and control of process
- Improve
 - Discipline
 - Software and systems engineering process capability
 - Product quality
 - Predictability/estimating capability
- CMMI places **proven approaches** into a structure that:
 - helps the organization **examine the effectiveness** of your processes
 - establishes **priorities for improvement**
 - help to **implement these improvements**

Improvements in CMMI

- Incorporates additional years of learning
- More explicitly **links best practices to business objectives**
- Adds more best practices (e.g.: Measurement, Risk Management, Product Integration, Decision Analysis and Resolution and Supplier Management)
- Captures more robust high-maturity practices
- Addresses additional generic practices needed for institutionalization
- Fully complies with relevant ISO standards

Example – low maturity

Organization

SoftComp

Project
Manager

John

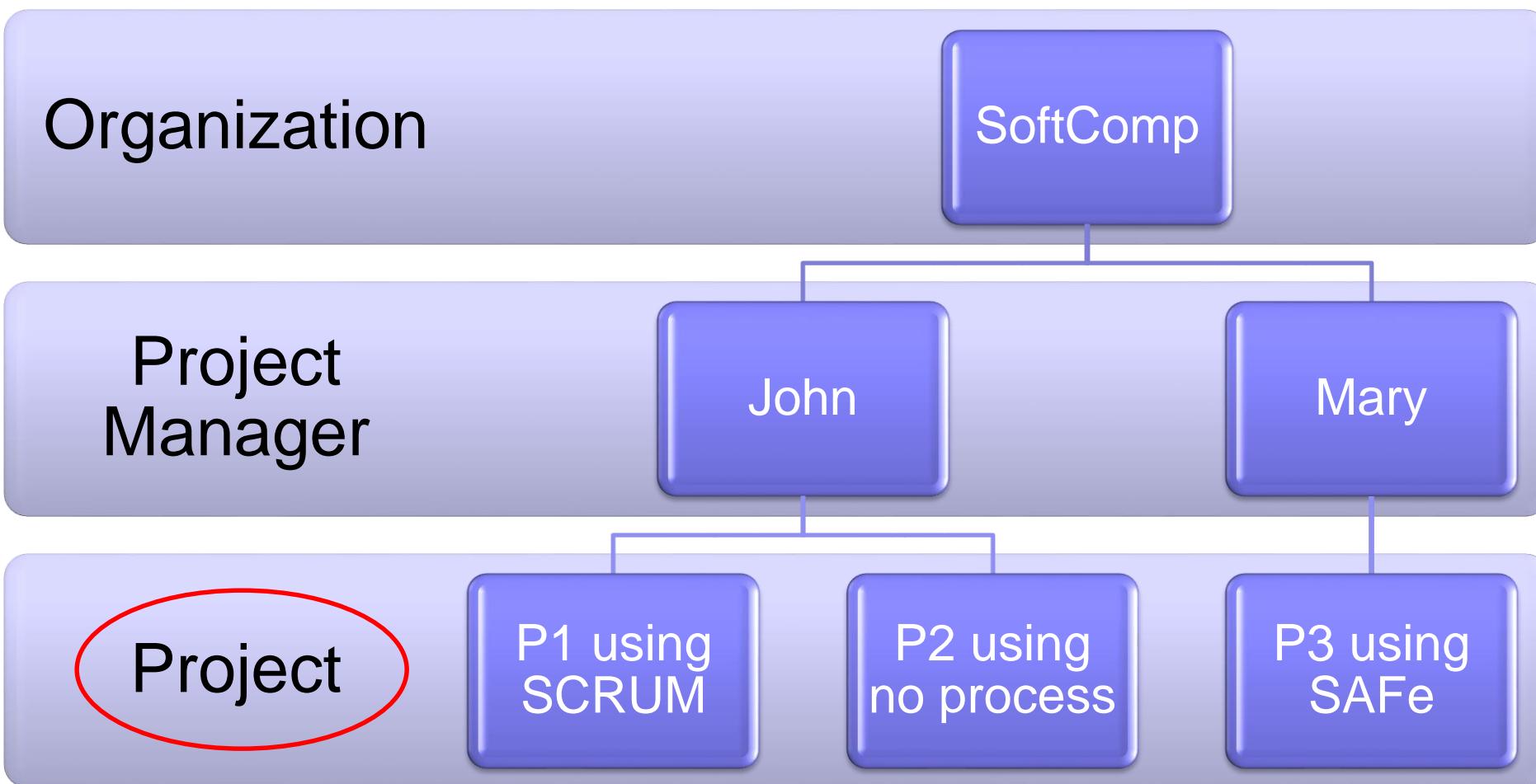
Mary

Project

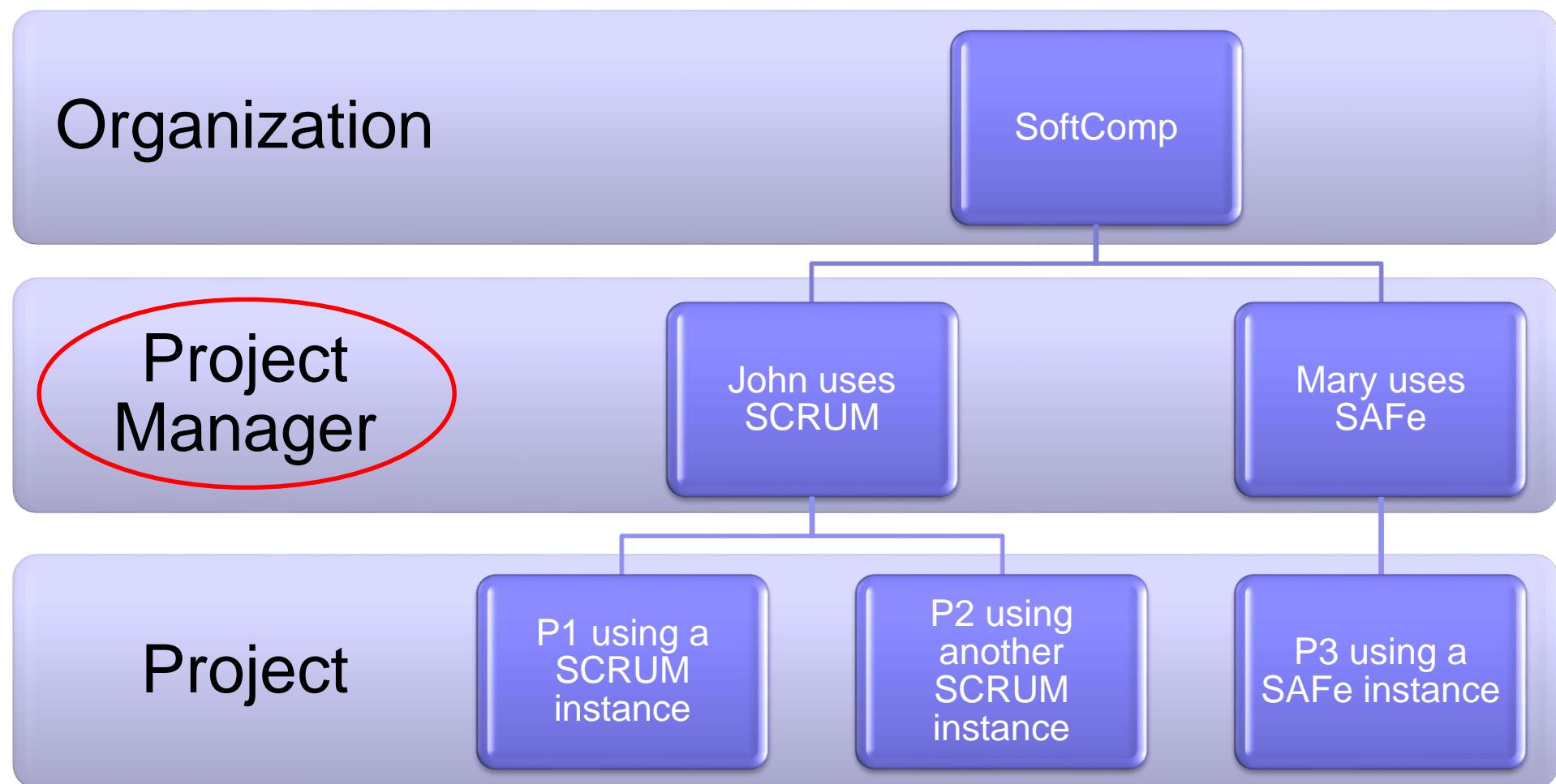
P1 using
SCRUM

P2 using
no process

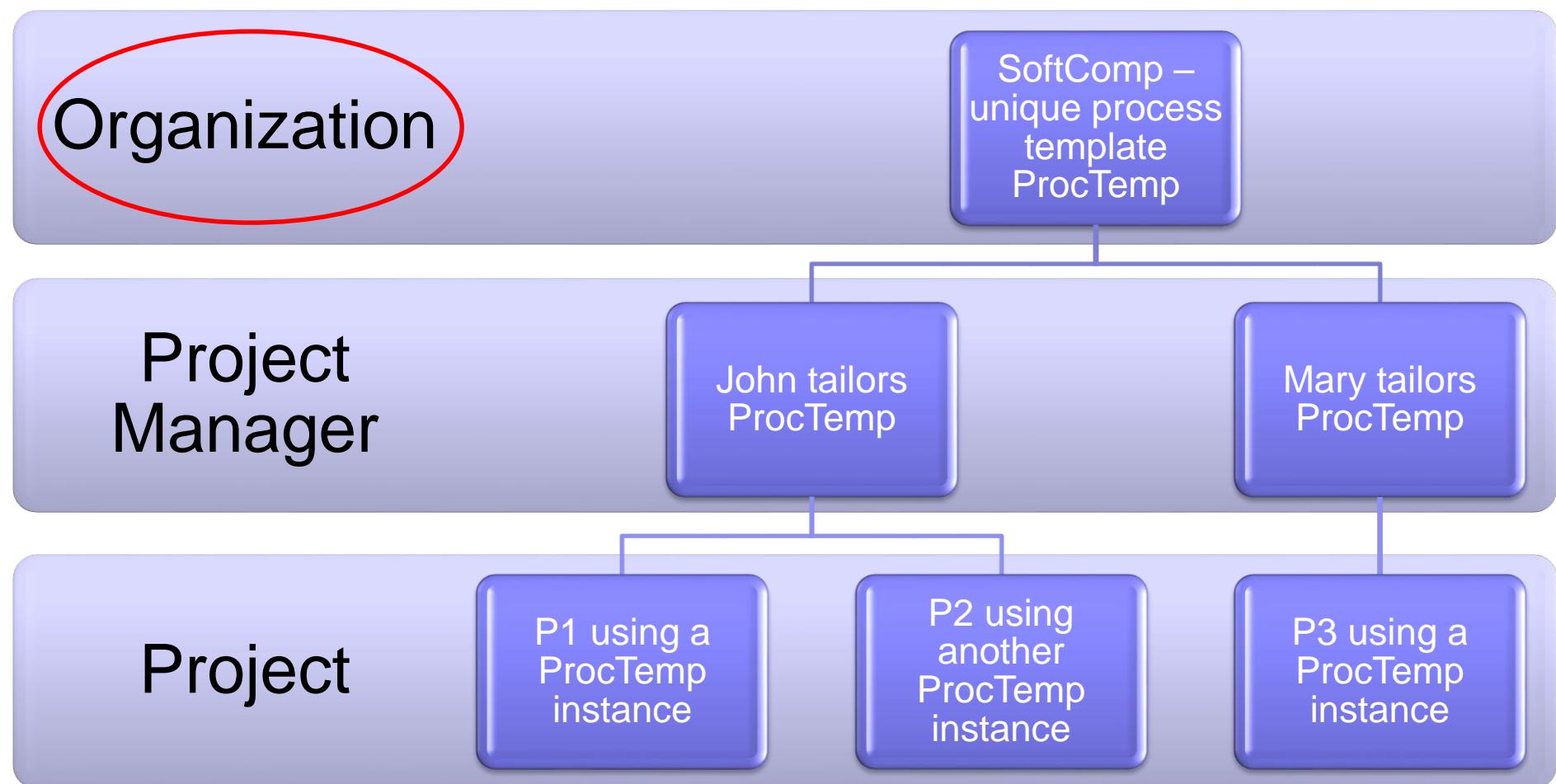
P3 using
SAFe

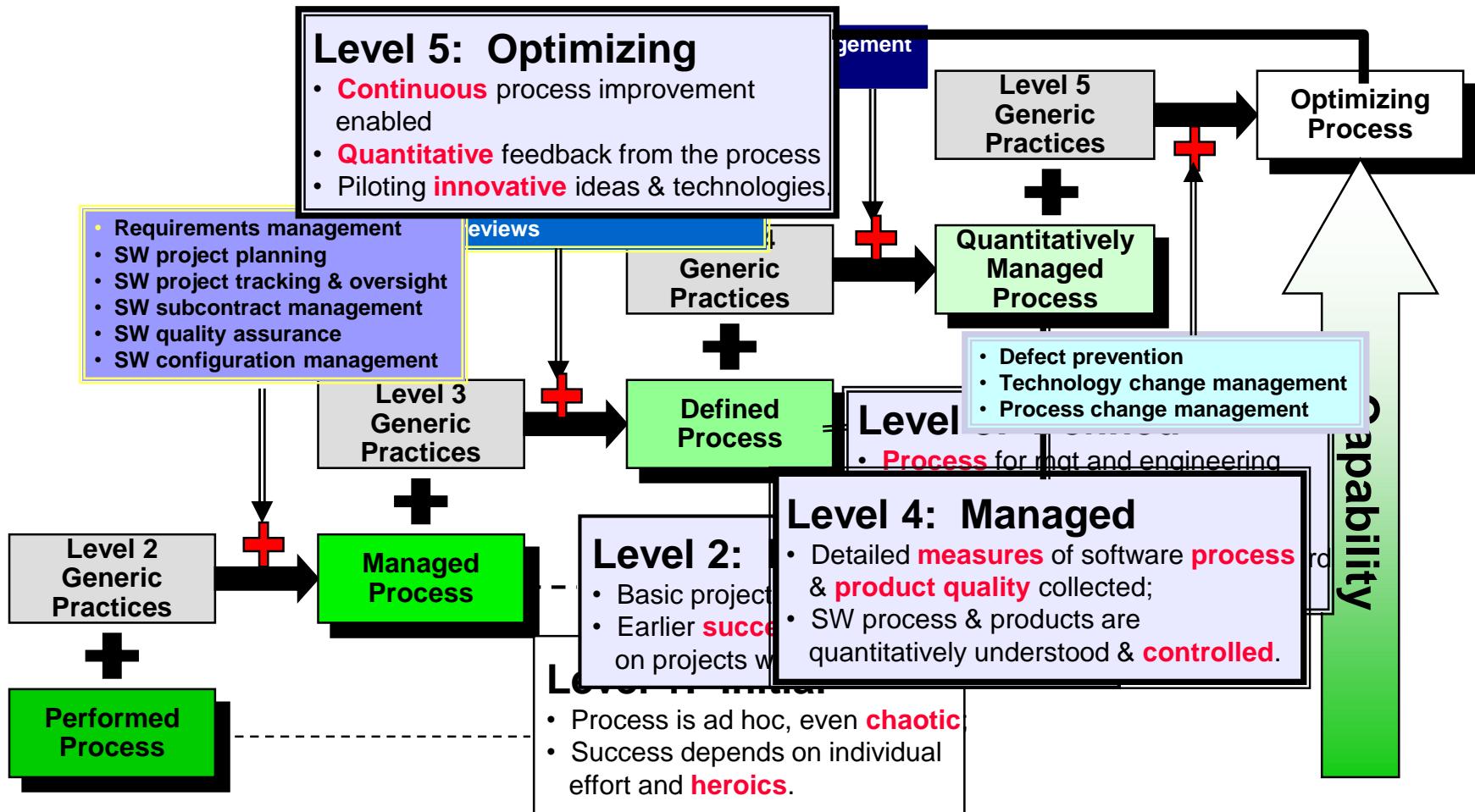


Example – higher maturity



Example – even higher maturity





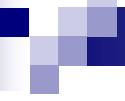
CMMI in Romania ?

Organization Organizational Unit	Team Leader Sponsor	Last Day of Appraisal Onsite	Model (Representation): Maturity Level
Automobile Dacia GROUP RENAULT Renault Services Romania	Tumu Satish Kumar Valentin Jipa	11/16/2018	CMMI-SVC v1.3(Staged):Maturity Level 2
Deloitte Tehnologie SRL Technology Projects	Rick Gibson Lucian Capdefier	05/23/2019	CMMI-DEV v1.3(Staged):Maturity Level 3
NTT DATA Romania Industries Business Unit and International Business Unit, Intercompany Division	Satoshi Yabe Daniel Metz	04/18/2018	CMMI-DEV v1.3(Staged):Maturity Level 2
Temenos India Private Limited Temenos UK Ltd Temenos USA Inc Temenos Ecuador SA Product Development Group T24 Development & Maintenance	Ramamurthy Rangarajan Suresh Bharathan S.P. Jayanthan	10/23/2019	CMMI-DEV v1.3(Staged):Maturity Level 3
Wind River Systems, Inc. Professional Services & Connected Vehicle	Adrian Gill Timothy Wietstock	05/17/2018	CMMI-DEV v1.3(Staged):Maturity Level 3

<https://sas.cmmiinstitute.com/pars/pars.aspx>

CMMI

- Process areas (PA)
- Project Management PA
 - Project Planning – PP
 - Project Monitoring and Control – PMC
- Related PAs
 - Requirements Management – RM
 - Configuration Management – CM
 - Risk Management – RM



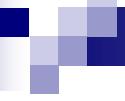
Project Planning Specific Goals

- SG1 – Establish Estimates
- SG2 – Develop a Project Plan
- SG3 – Obtain Commitment to Plan

[CMMI definition]

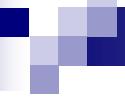
PP SG1 – Establish Estimates

- SP 1.1 Estimate the Scope of the Project
- SP 1.2 Establish Estimates of Work Product and Task Attributes
- SP 1.3 Define Project Lifecycle
- SP 1.4 Determine Estimates of Effort and Cost



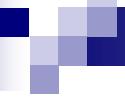
PP SG2 – Develop a Project Plan

- SP 2.1 Establish the Budget and Schedule
- SP 2.2 Identify Project Risks
- SP 2.3 Plan for Data Management
- SP 2.4 Plan for Project Resources
- SP 2.5 Plan for Needed Knowledge and Skills
- SP 2.6 Plan Stakeholder Involvement
- SP 2.7 Establish the Project Plan



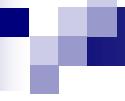
PP SG3 – Obtain Commitment to Plan

- SP 3.1 Review Plans That Affect the Project
- SP 3.2 Reconcile Work and Resource Levels
- SP 3.3 Obtain Plan Commitment



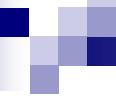
Project Monitoring and Control Specific Goals

- SG 1 Monitor Project Against Plan
- SG 2 Manage Corrective Action to Closure



PMC SG1 – Monitor Against Plan

- SP 1.1 Monitor Project Planning Parameters
- SP 1.2 Monitor Commitments
- SP 1.3 Monitor Project Risks
- SP 1.4 Monitor Data Management
- SP 1.5 Monitor Stakeholder Involvement
- SP 1.6 Conduct Progress Reviews
- SP 1.7 Conduct Milestone Reviews

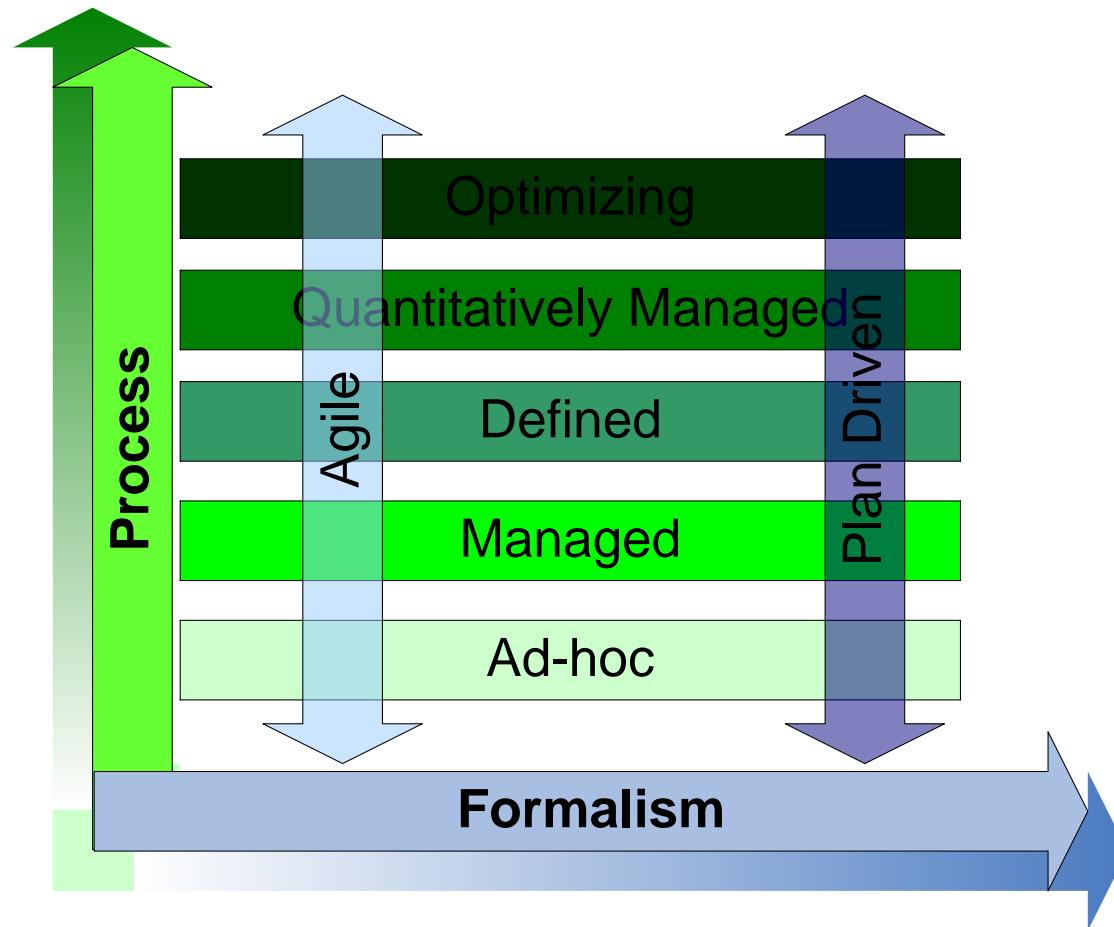


PMC SG2 – Manage Corrective Action to Closure

- SP 2.1 Analyze Issues
- SP 2.2 Take Corrective Action
- SP 2.3 Manage Corrective Action

Process levels and methodologies

- Mature processes => plan driven methodologies?
- Can we apply agile methods in mature processes?



Methodologies compliance with CMMI - Gap Analysis

- Methodologies measure with SE process compliance employing gap analysis
- Developing a gap analysis is a long and tedious effort
- Software automation tools can be used (ex. CMMILive)

Gap Analysis Example

Goal	Practice	PRJ	Evidence			
			Direct	Direct Hyperlink	Indirect	Indirect Hyperlink
SG 1	SP 1.1 Establish and maintain the project's defined process from project startup through life of project.	Org	DSP - USA007683 <i>Section 4.2.1.4 identifies that the project lead will determine the process (describes how) for the project using the DSP and any applicable tailoring or</i>	DSP		
		All	SDP (Document) <i>Section 1.2 "Approach" describes the project's process and points to SDP section 8.0 for any applicable tailoring and any process waivers. Section 1.2 also specifies the project's defined software process as the TIS DSP with any tailoring or process waivers identified in Section 8.0.</i> <i>PPD Note : There is no tailoring of the organization's defined process specific to PPD.</i>		SDP Revision log and approval (formulate) <i>Sign off on the SDP shows approval of the project's defined process.</i>	
		All	Project Schedule (use) <i>The process requires the use of a project schedule. Link is to the TIS Schedule Template.</i>			
	SP 1.2 Use organizational process assets and measurement repository for estimating and planning the project's activities.	ORG	DSP - USA007683 <i>Section 4.2.1.9 directs the use of past actuals for estimation and the use organizational process assets in developing the plan and other work products.</i>	DSP		
		All	SDP (document) <i>Software Development Plan for the project is based on the SDP Template and previous projects.</i>		SDP approval (formulate) <i>Sign off on the SDP shows approval of the project's estimating and planning process. Summarized estimated effort in Section 5.4 "Staffing requirements /Plan".</i>	
		All	TIS Estimation Tool (use) <i>The estimation tool uses standard values calibrated using historical data. The SDP directs the use of the TIS Estimation Tool by the Project Lead to produce the project Estimation Worksheet (project specific effort estimate).</i> <i>Log shows updates to estimation values in the tool. When Project Leads modify values calculated in the TIS Estimation Tool, a reference number is entered in the "BOE" column in the Estimation worksheet, and a rationale is placed in the BOE Tab worksheet.</i>			

Gap Analysis Results Example

Who's Resp	Chris	Chris	Chris	Chris	Donna	Donna	Donna	Chris	Chris	Chris	Chris	Chris	Chris	Chris	Chris	Chris	Chris	Chris	Chris	Chris
Process Areas	Engr	Proj Mgmt	Proj Mgmt	Proj Mgmt	Support	Support	Support	Engr	Engr	Engr	Engr	Engr	Process Mgmt	Process Mgmt	Process Mgmt	Proj Mgmt	Proj Mgmt	Support		
Maturity Level	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3
Process Areas	REQM	PP	PMC	SAM	MA	PPQA	CM	RD	TS	PI	VER	VAL	OPF	OPD	OT	IPM	RSKM	DAR		
SG1																				
SP 1.1	S	S	S		S	S	S	S	P	P	P	S	S	S	S	S	S	S		
SP 1.2	S	S	S		S	S	S	P	S	P	S	S	S	S	S	S	S	S	S	
SP 1.3	P	S	S		S		P		P	S	S	S	S	S	S	S	S	P		
SP 1.4	S	S	S		S										S	S	S		P	
SP 1.5	P		S											S		S			P	
SP 1.6			S																P	
SP 1.7			S																	
SG2																				
SP 2.1		S	S		S	S	P	S	S	P	S	P	S		S	S	S			
SP 2.2		S	S		S	P	S	S	S	P	P	S	S		S	S	S			
SP 2.3		S	S		P			P	S		S		S		S	S	S			
SP 2.4		S			P				S			S			S					
SP 2.5		S																		
SP 2.6		S																		
SP 2.7		S																		
SG3																				
SP 3.1		S				P	S	S	P	S					S	S				
SP 3.2		S				P	S	S	P	P					S	S	S			
SP 3.3		S					P		S											
SP 3.4							S		S											
SP 3.5							P													

Range from CMMI

Process approach vs. PM role

■ Ad-hoc

- Not documented and depends on involved parties
- PM has absolute control

■ Process–driven

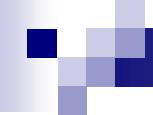
- Documented processes on all activities
- Organizational infrastructure enables successful execution of projects

Ad-hoc approach

- Dynamic environment
- Leader has absolute control
- Fast response to environmental changes
- Can be profitable
- Perfect for pinning the blame on one person
- Reduces process overhead
- Creates uncertainty
- Centralizes authority
- Unpredictable results
- Focuses on people monitoring
- Organizational bandwidth depends on leaders

Works good for...

- Small organization
- Small number of PMs in organization (i.e. 2-3)
- Small number of concurrent projects (< 5)



Process-driven approach

- Minimizes person dependency
- Enables good performance for beginners
- Facilitates integration of project experience into process
- Monitors projects rather than people
- Involves the organization
- Provides uniformity
- Facilitates measurement
- Builds basis for predictability

Right choice?

- Financial processes (enforced by law)
- HR processes (enforced by law, fairness)
- Project management processes (no fairness/law enforcement!)

LET'S SCHEDULE A SCENARIO-BASED ROUNDTABLE DISCUSSION ABOUT OUR ENTERPRISE PROJECT MANAGEMENT.



Dilbert.com DilbertCartoonist@gmail.com

WE'LL USE OUR INFRASTRUCTURE SURVEY TOOL TO ARCHITECT A RISK-BASED TIERING SYSTEM.



i-910 ©2010 Scott Adams, Inc./Dist. by UFS, Inc.

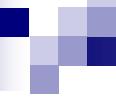
THAT ALMOST MEANT SOMETHING.



I'M TEMPTED TO STOP ACTING RANDOMLY.

Process-driven but...

- What process?
 - ISO
 - CMMI
- How much process?



Software Development Methodologies

- Plan-driven
- Agile

Plan based - UP Principles

- Most prominent example is RUP
- Essentials of the UP (© RUP)
 - Develop a **Vision**
 - Manage to the **Plan**
 - Mitigate **Risks** and Track Related Issues
 - Examine the **Business Case**
 - Design a **Component Architecture**
 - **Incrementally** Build and Test the Product
 - Regularly Assess **Results**
 - Manage and Control **Changes**
 - Deploy a **Usable Product**
 - Adopt a **Process** that Fits Your Project

Project Management Definition (according to RUP)

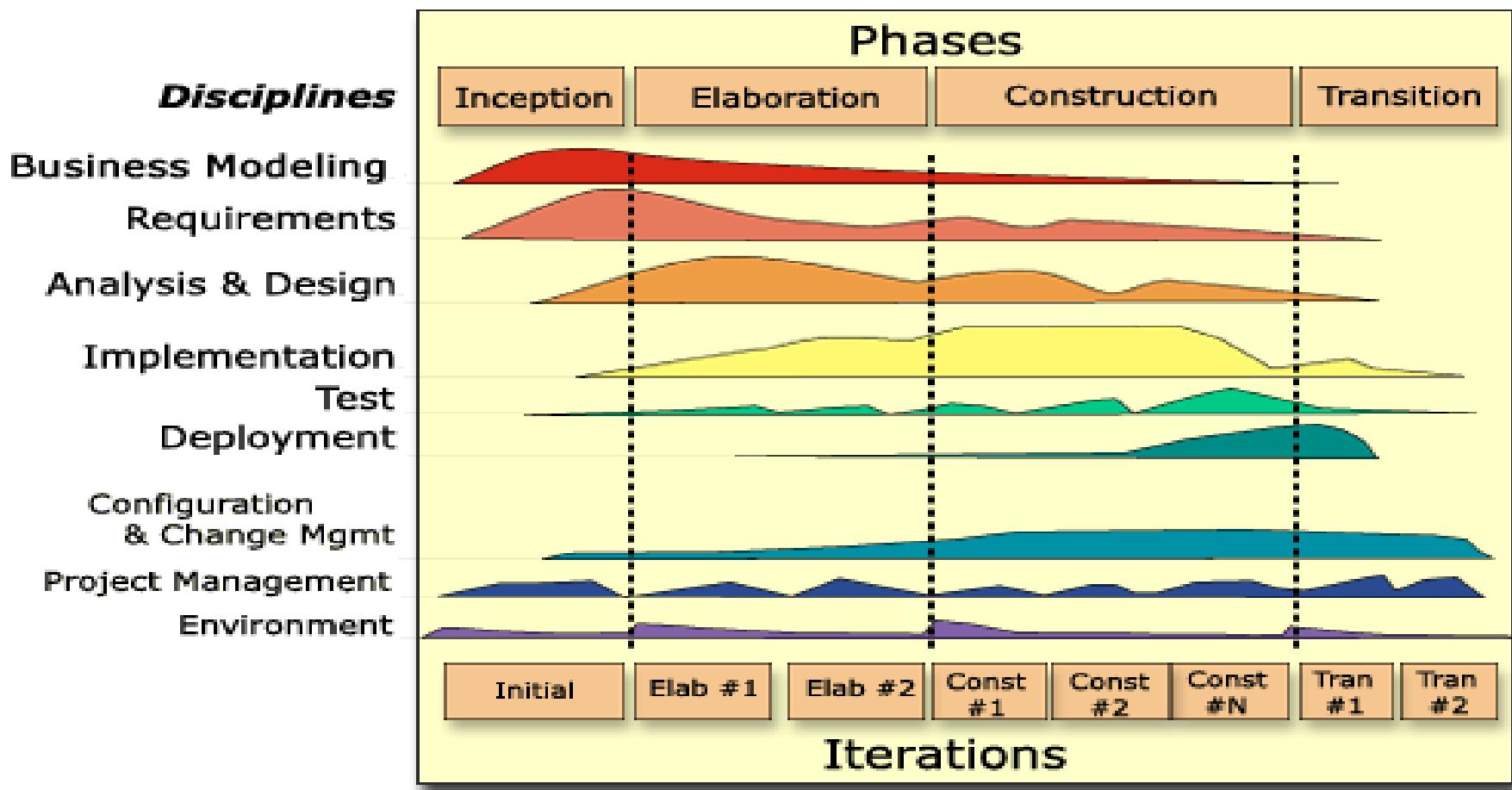
“Software Project Management is the art of

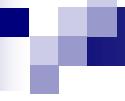
- balancing competing objectives,
- managing risk,
- overcoming constraints

to deliver, successfully, a product which meets the needs of both customers (the payers of bills) and the users.

The fact that so few projects are unarguably successful is comment enough on the difficulty of the task.”

The Unified Process





Agile Methodologies Principles

- ... how are they different?

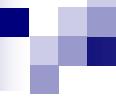
Plan-driven vs. Agile

■ Common features

- Commitment to quality
- Communication
- Iterative

■ Difference

- Level of formalism
- Time-span for planning



Agile Values

- People and interactions are more important than development tools
- Working software is more important than complete documentation
- Collaboration with clients is more important than contract negotiation
- Response time to problems is more important than following a plan

Agile Methodologies Principles (1) –

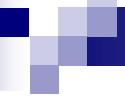
<http://www.agilemanifesto.org/principles.html>

- An agile, scalable approach to requirements management, development and system testing
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

Agile Methodologies Principles (2) –

<http://www.agilemanifesto.org/principles.html>

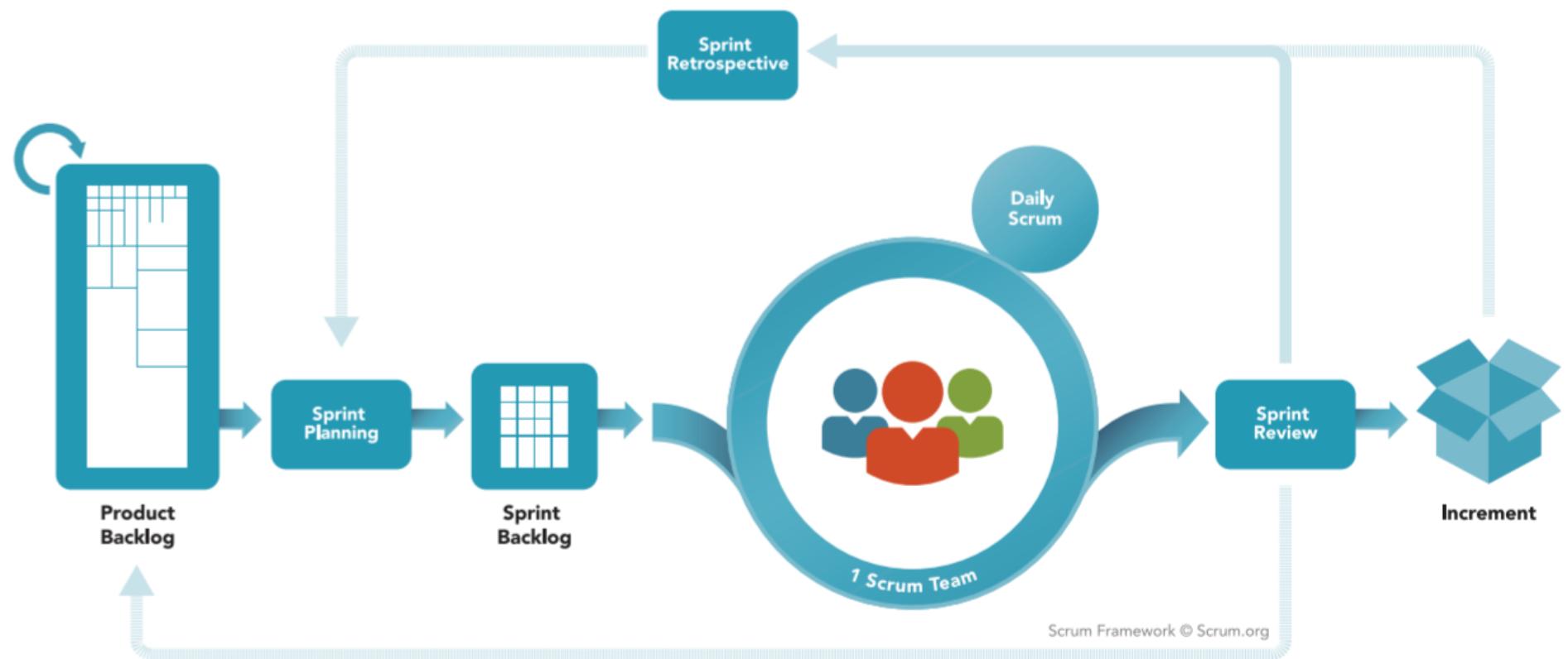
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity - the art of maximizing the amount of work not done - is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



Agile Methodologies

- Scrum
- XP
- Lean
- SAFe
- Others....

SCRUM

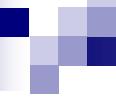


SCRUM Team

- Self-organizing and cross-functional
- Consists of
 - Product Owner
 - Development Team
 - Scrum Master

Product Owner

- Responsible for maximizing the value of the product resulting from work of the Development Team
- Manages Backlog by
 - Clearly expressing Product Backlog items;
 - Ordering the items in the Product Backlog
 - Optimizing the value of the work of the Development Team
 - Ensuring that the Product Backlog is visible, transparent, and clear to all
 - Ensuring the Development Team understands items in the Product Backlog to the level needed.



Development Team

- self-organizing.
- cross-functional,
- no titles for Development Team members
- no sub-teams in the Development Team, regardless of domains that need to be addressed like testing, architecture, operations, or business analysis; and,
- individual Development Team members may have specialized skills, but accountability belongs to the Development Team as a whole.

SCRUM Master

- is a servant-leader for the Scrum Team
- Serves the Product Owner
 - Ensuring that goals, scope, and product domain are understood
 - Finding techniques for effective Product Backlog management;
 - Understanding product planning in an empirical environment;
 - Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value;
 - Understanding and practicing agility; and,
 - Facilitating Scrum events as requested or needed.

SCRUM Master

- Serves the Development Team
 - Coaching the Development Team in self-organization and cross-functionality;
 - Helping the Development Team to create high-value products;
 - Removing impediments to the Development Team's progress;
 - Facilitating Scrum events as requested or needed; and,
 - Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.

SCRUM Master

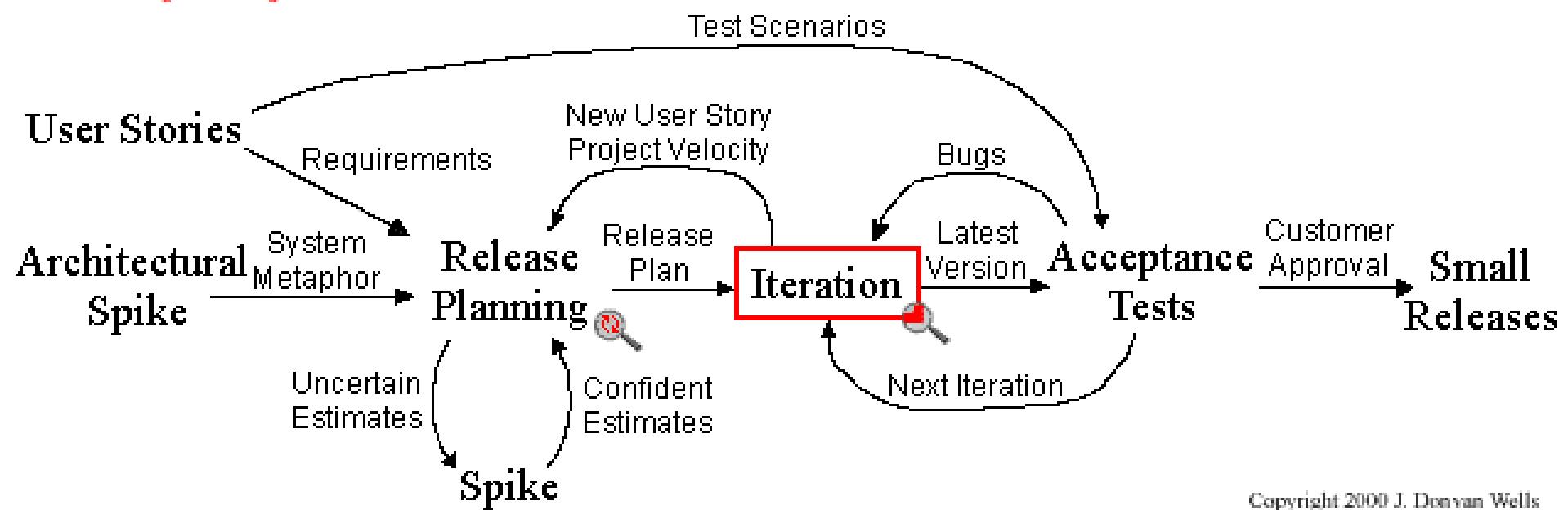
■ Serves the Organization

- Leading and coaching the organization in its Scrum adoption;
- Planning Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact Scrum and empirical product development;
- Causing change that increases the productivity of the Scrum Team; and,
- Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.

XP



Extreme Programming Project

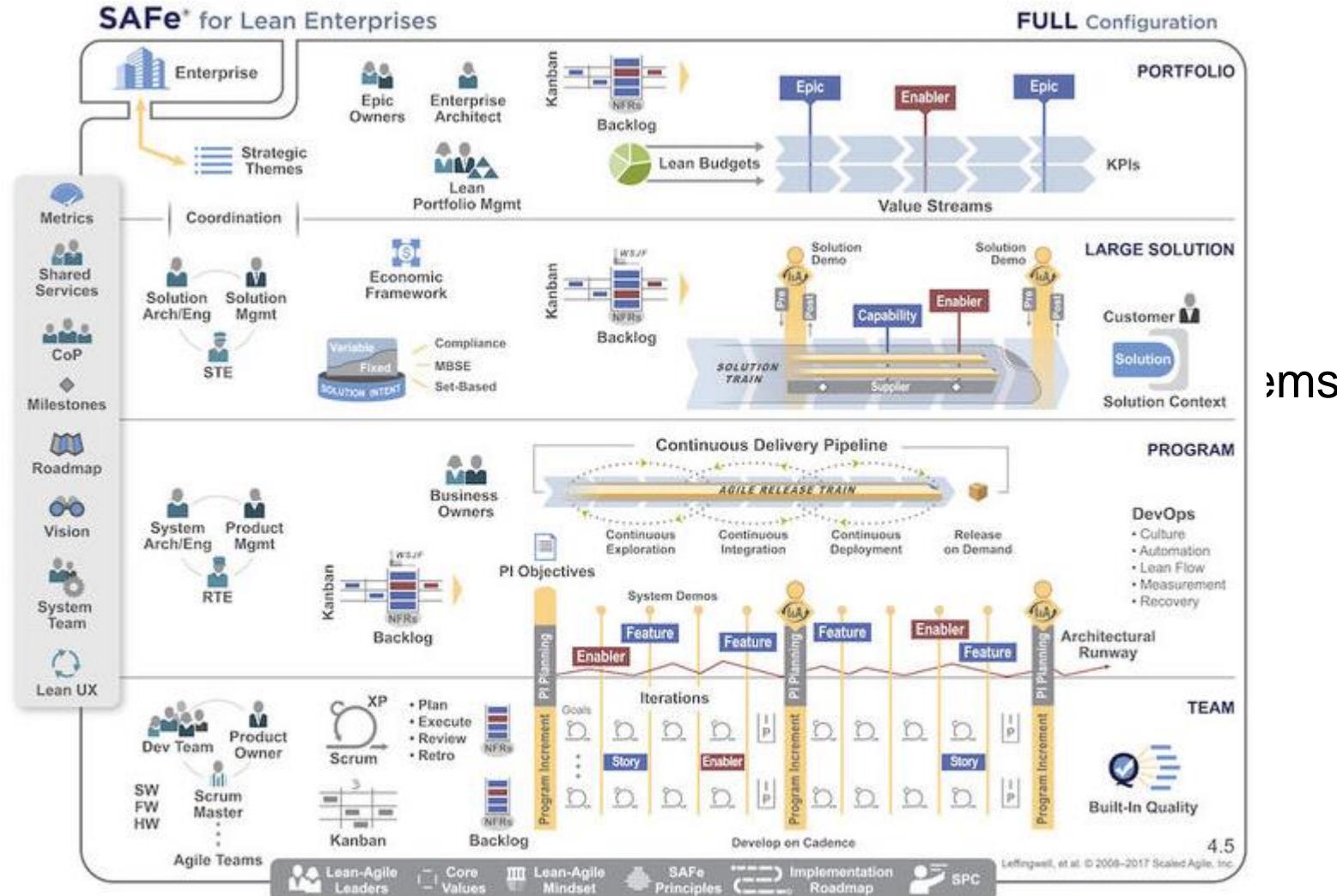


Copyright 2000 J. Don van Wells

Lean

- Eliminating Waste
- Amplifying Learning
- Deciding as Late as Possible
- Delivering as Fast as Possible
- Empowering the Team
- Building Integrity In
- Seeing the Whole

Scaled Agile Framework (SAFe)

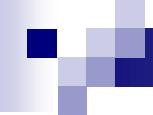


Additional links

- <https://cmmiinstitute.com/resources>
- [How to Fail with the Rational Unified Process: Seven Steps to Pain and Suffering](#)
- SAFE <https://www.ivarjacobson.com/scaled-agile-framework>

Wrap-up

- CMMI measures the level of process
- Plan-driven vs. Agile
 - time-span for planning
 - degree of formalism
- PM has to decide the Process level and degree of formalism for the specific project

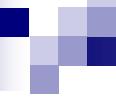


Quiz time

- Switch over to Moodle

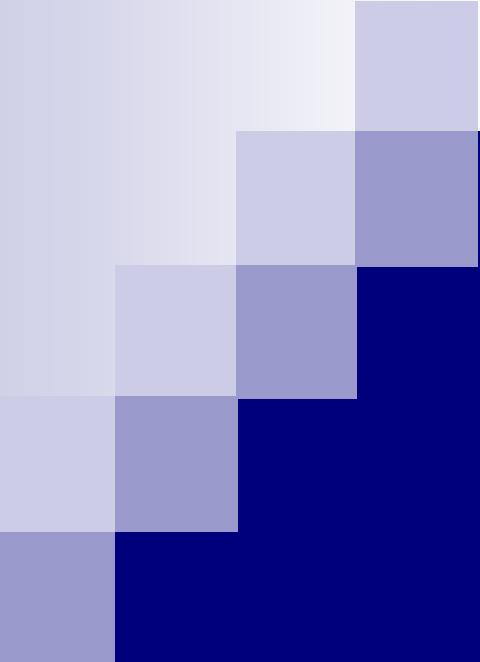
Discussion topic next week

- Here you will find "Value Agile" by Tom Gilb, 2020.
 - Choose one agile principle that is discussed and reformulated by Gilb and present it.
- OR
- Present your own critical opinion about that principle.



Next Meeting

- Introduction to Project Management part II – PM Techniques

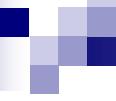


Project Management Techniques

Lecture 3

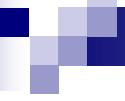
Topics

- Project Management Activities and Techniques
- PM in Plan-driven (RUP)
- PM in Agile



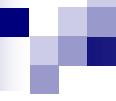
What is a project

- A **unique** venture with specific **timeframe**
- Conducted by **people**, often from various backgrounds
- **Goal** oriented, clear target
- **Constrained** by cost, schedule, resources, functionality and quality
- **Divisible** into phases and iterations
- Has a clear **priority** in the organization



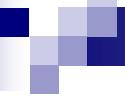
Project Management Skills

- Techniques
- Tools
- People management



Techniques

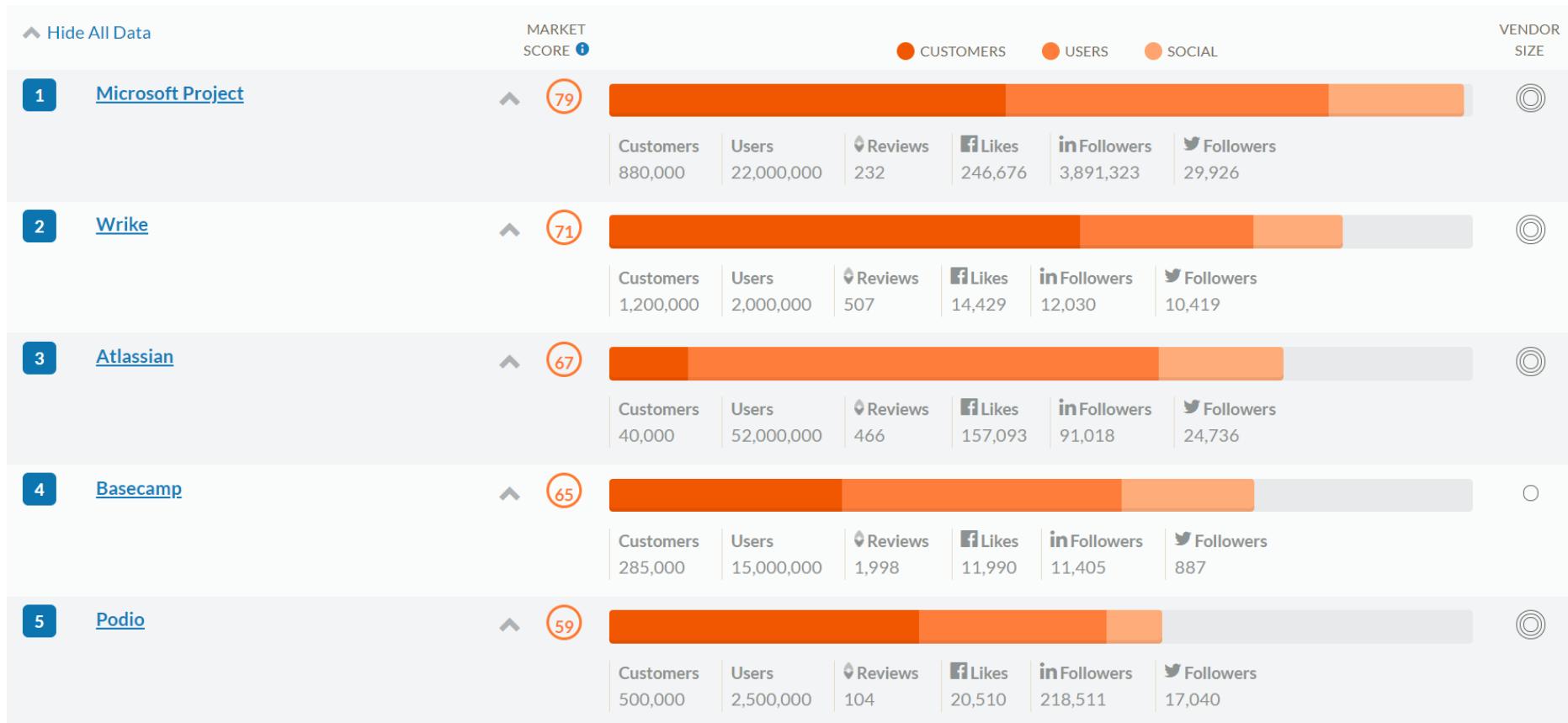
- Planning the project
- Implementation of the project plan
- Tracking, reporting, and control of projects as they progress
- Project closure and completion of the project



Tools

- Planning
- Tracking
- Reporting

Project Management Software



<https://www.capterra.com/project-management-software/#infographic>

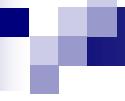
PM Software

1. Wrike
2. Smartsheet
3. Clarizen
4. Workzone
5. Hive
6. Backlog
7. Celoxis
8. Mavenlink
9. Jira
10. Proworkflow

Product	Category	User Rating	Pricing Tier	Mobile App	Free Trial
 Trello +	Agile	4.36/5	Low-end		
 Basecamp +	General Use	4.01/5	Average		
 Asana +	General Use	4.21/5	Low-end		
 CA Clarity +	Enterprise	3.74/5	Average		
 Atlassian JIRA +	Agile	4.2/5	Average		
 Microsoft Project +	Enterprise	4/5	High-end		
 Huddle +	General Use	4.37/5	High-end		
 Podio +	General Use	4.12/5	Low-end		
 LiquidPlanner +	Agile	4.4/5	Average		
 Wrike +	Enterprise	4.34/5	Average		

<https://project-management.com/top-10-project-management-software/>

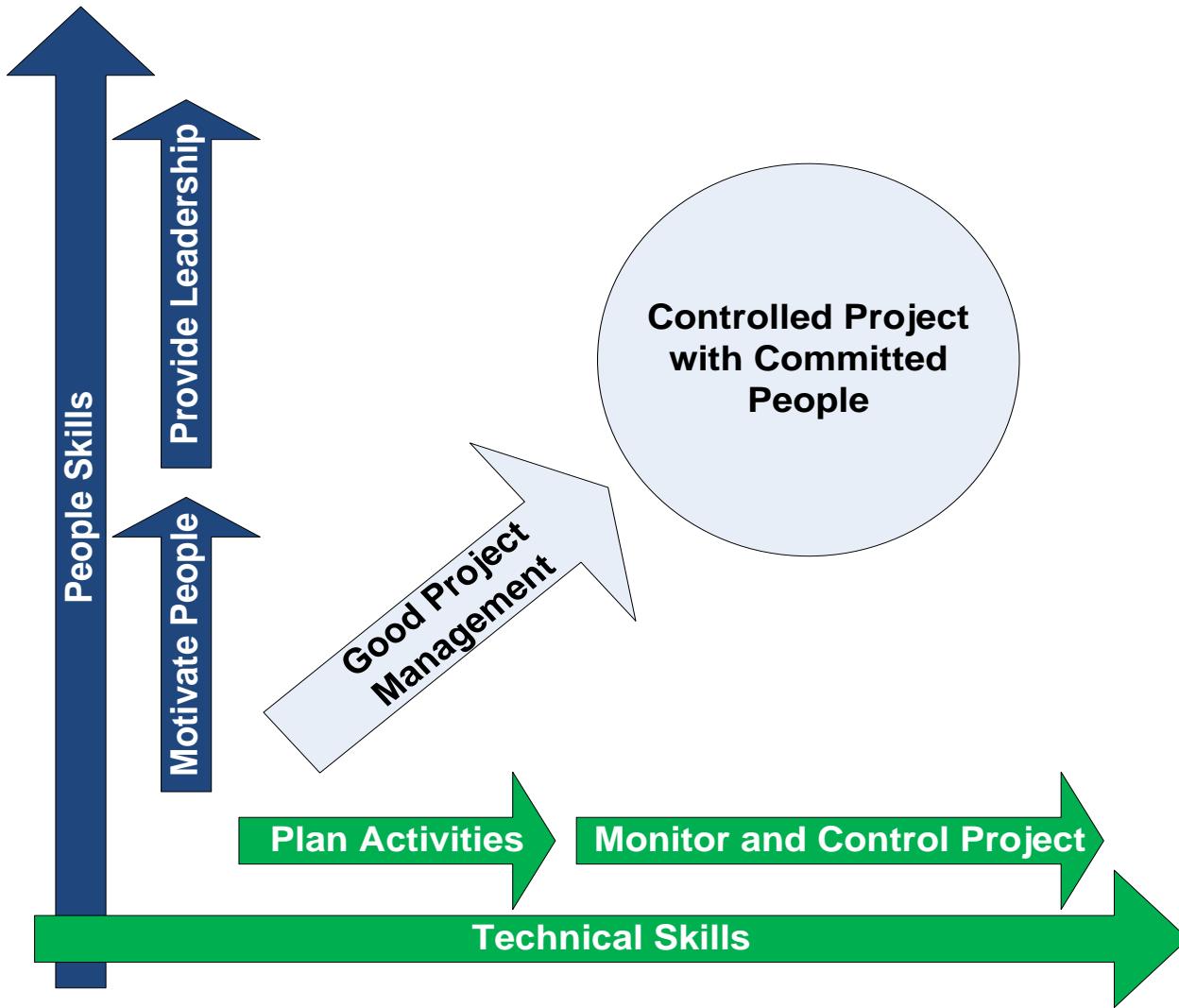
<https://technologyadvice.com/project-management>



People

- Manage people who implement the project
- Manage stakeholder expectations
- Lead

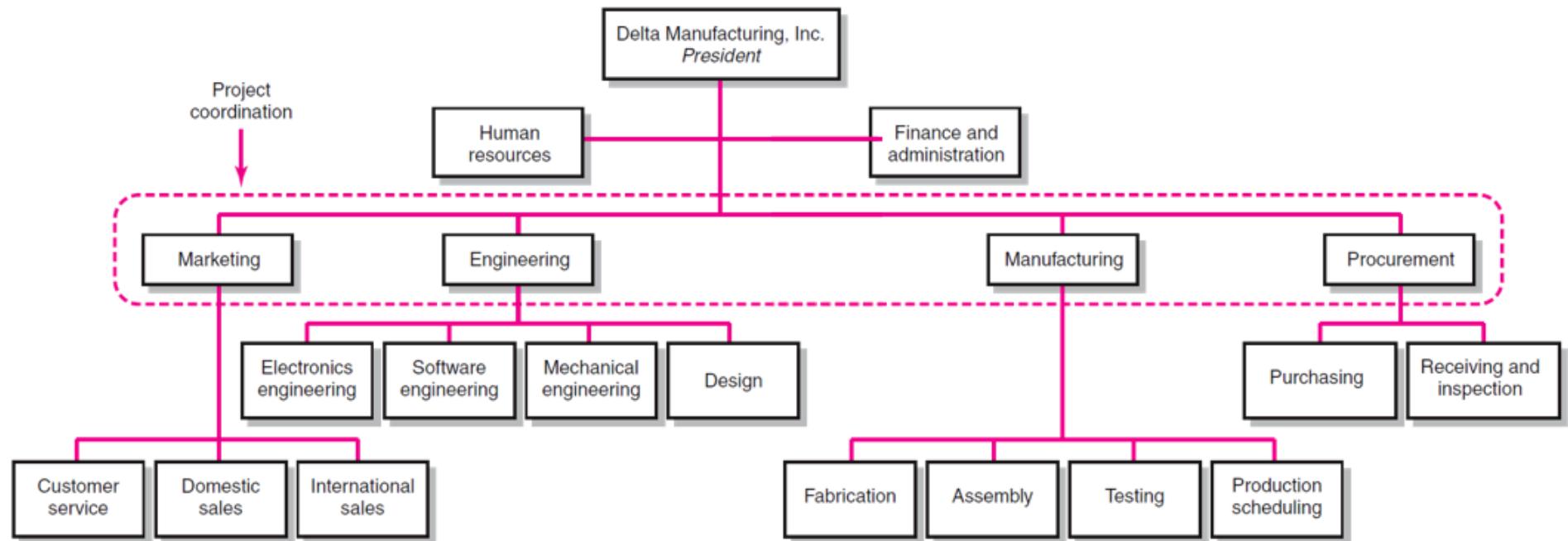
PM Skill set



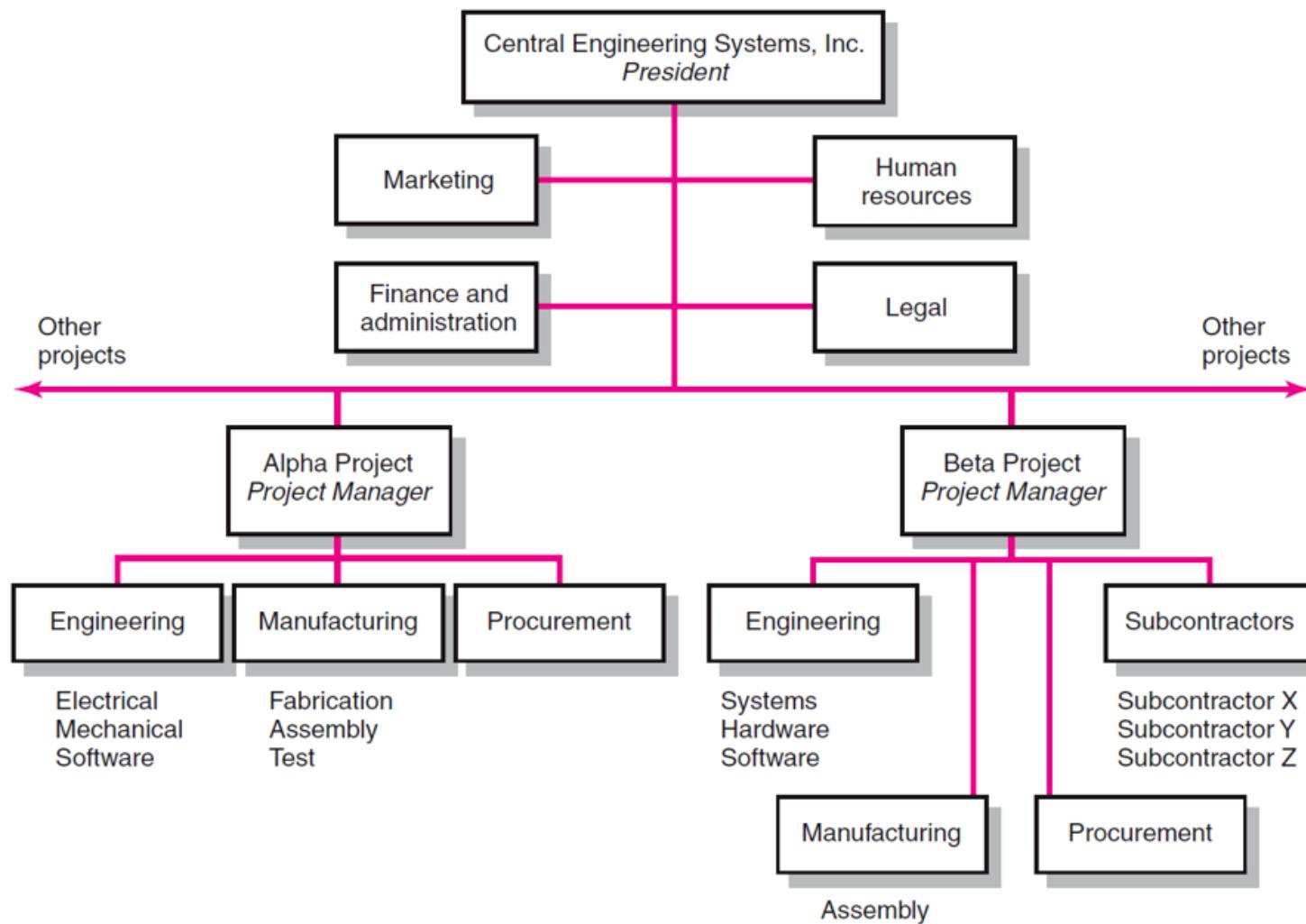
Organizational structure

Project Characteristics	Organization Structure <i>Functional</i>	<i>Matrix</i>			<i>Projectized</i>
		Weak Matrix	Balanced Matrix	Strong Matrix	
Project Manager's Authority	Little or None	Limited	Low to Moderate	Moderate to High	High to Almost Total
Resource Availability	Little or None	Limited	Low to Moderate	Moderate to High	High to Almost Total
Who controls the project budget	Functional Manager	Functional Manager	Mixed	Project Manager	Project Manager
Project Manager's Role	Part-time	Part-time	Full-time	Full-time	Full-time
Project Management Administrative Staff	Part-time	Part-time	Part-time	Full-time	Full-time

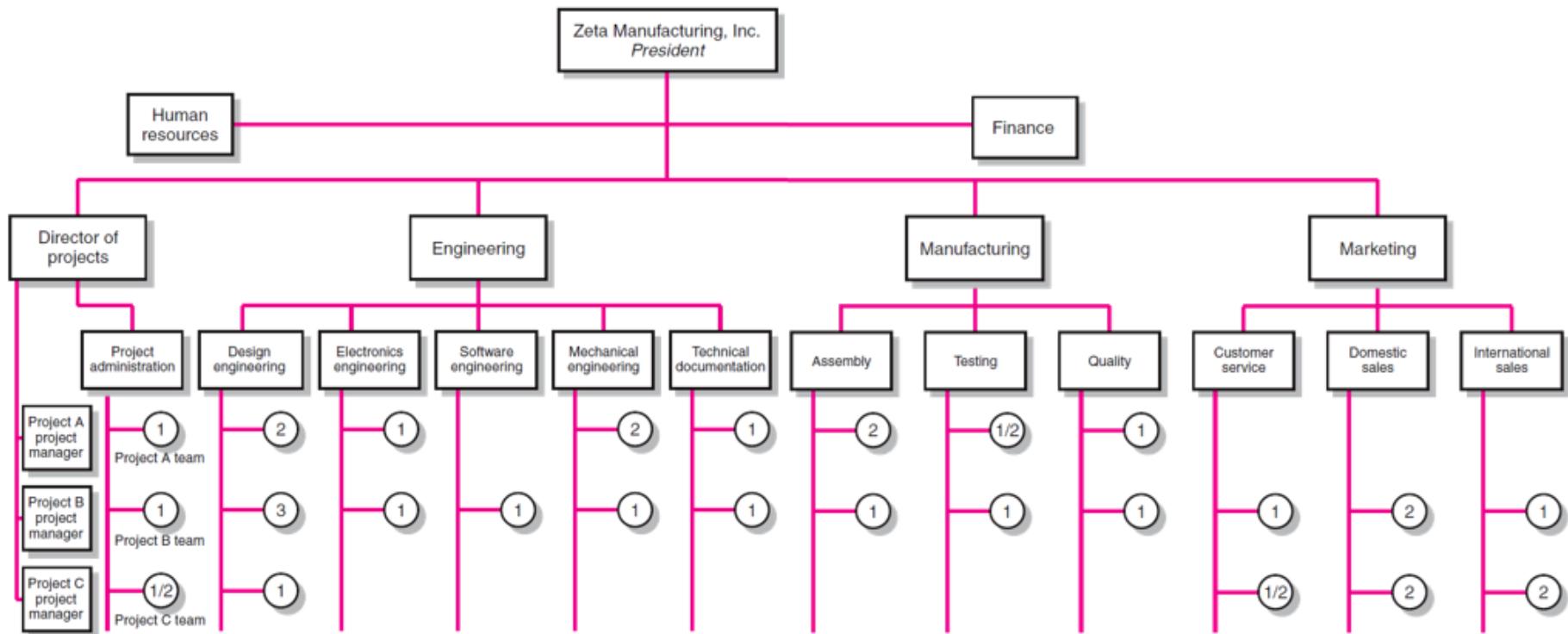
Functional organization



Projectized organizational structure



Matrix organizational structure



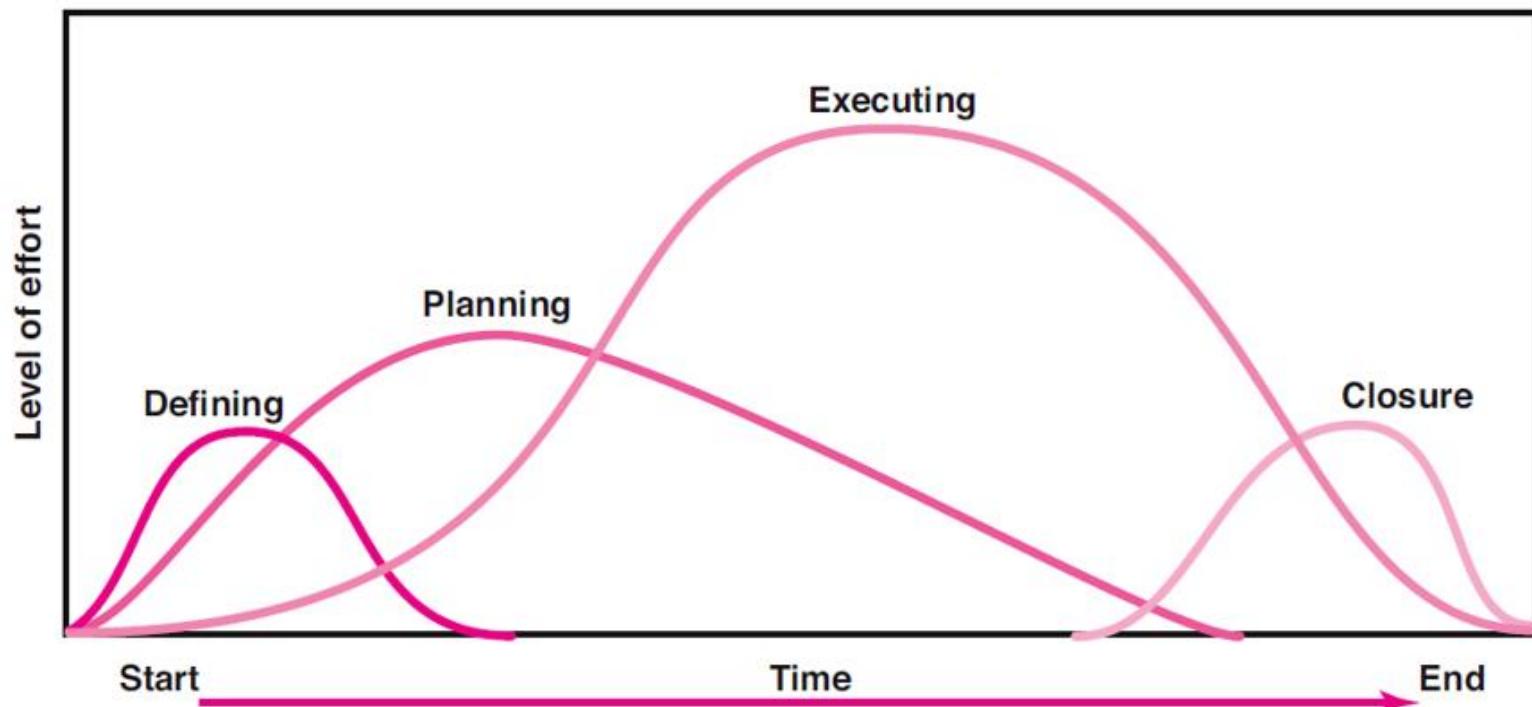
Division of responsibilities in matrix

Project Manager	Negotiated Issues	Functional Manager
What has to be done?	Who will do the task?	How will it be done?
When should the task be done?	Where will the task be done?	
How much money is available to do the task?	Why will the task be done?	How will the project involvement impact normal functional activities?
How well has the total project been done?	Is the task satisfactorily completed?	How well has the functional input been integrated?

Project Management

- How do we manage a project?
- Where do we start?
- How can we make sure that we'll be successful?
- How can we avoid failure?
- How can we balance control and encourage creativity?

Project Life Cycle



Defining

1. Goals
2. Specifications
3. Tasks
4. Responsibilities

Planning

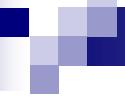
1. Schedules
2. Budgets
3. Resources
4. Risks
5. Staffing

Executing

1. Status reports
2. Changes
3. Quality
4. Forecasts

Closure

1. Train customer
2. Transfer documents
3. Release resources
4. Evaluation
5. Lessons learned

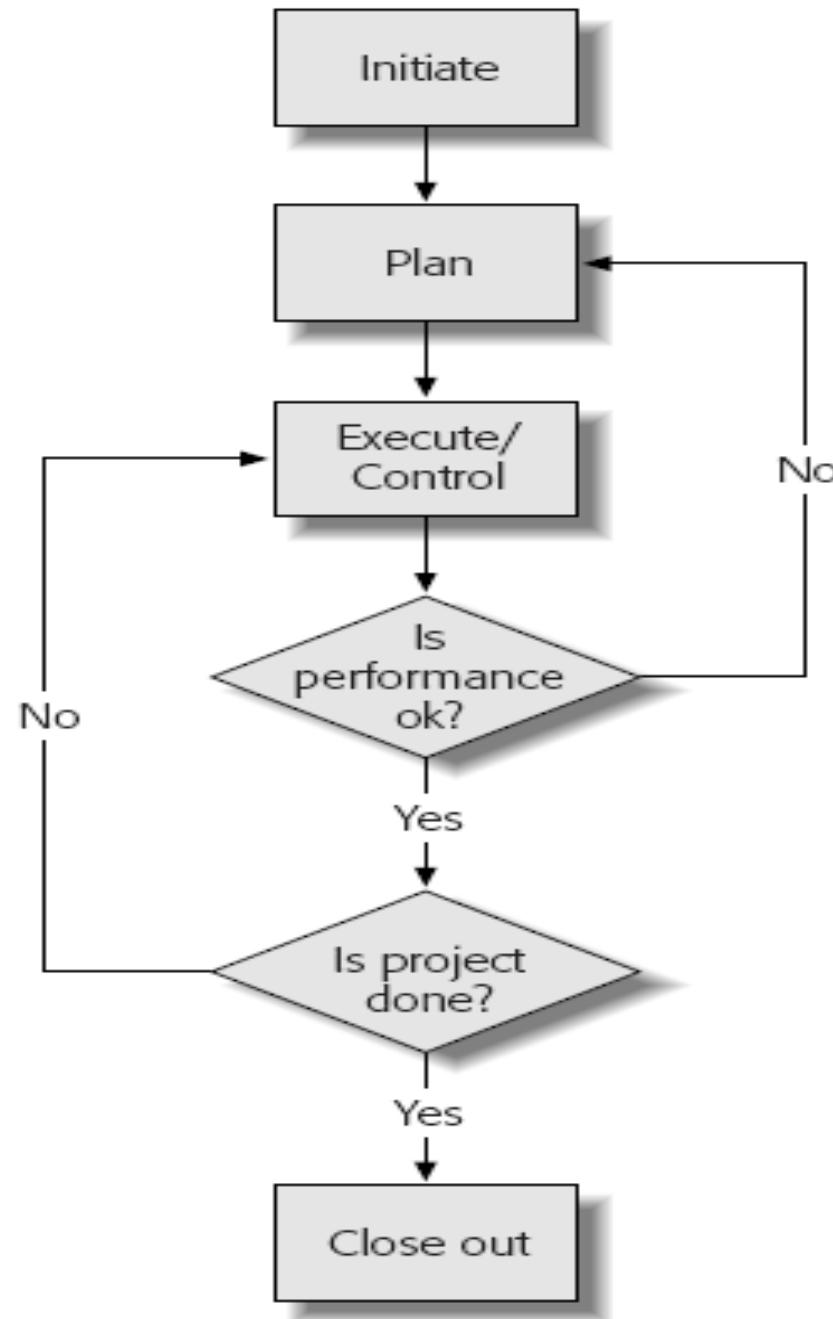


Project Management Processes (PMBOK)

- Scope
- Integration
- Time
- Cost
- Quality
- Human Resources
- Communication
- Risk
- Procurement

PM process types

- Project Management discipline perspective
 - Core processes
 - Support processes
- Core Project Management perspective
 - Initiation Processes
 - Planning Processes
 - Executing Processes
 - Controlling Processes
 - Closing Processes



Initiation

■ Why are we going to perform this project?

- Problem statement
- Project mission

■ What are we going to achieve?

- Project objectives

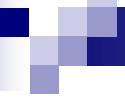
Problem statement

- We need a TV commercial to advertise our products.



- Sales have dropped 40 percent in the last year, and a customer survey identifies poor quality as the primary reason. Our investors are getting nervous.

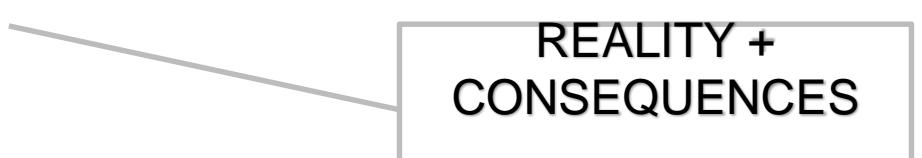
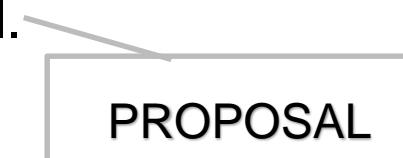




Problem statement elements

- **Ideal:** identify the ideal situation
- **Reality:** identify current situation => describe gaps
- **Consequences:** state the consequences of the problem
- **Proposal:** propose a solution for the problem

Example

- Remote workers across the company should be able to communicate with one another seamlessly and effortlessly, without getting bogged down in unnecessary or irrelevant messages.
IDEAL
- Right now, messages tend to get lost or cluttered through multiple email strings. This results not only in lost productivity, but also multiple problems in miscommunication.
REALITY + CONSEQUENCES
- I propose that all employees sign up for Slack and use it for the majority of in-company communication. Conversations can be easily searched and organized by channel. Only more formal communication, like official announcements, should be sent over email.
PROPOSAL

Project mission

“You’ve got to be very careful if you don’t know where you’re going, because you might not get there.”
[Yogi Berra]

- Who is the customer?
- What is the project supposed to accomplish?
- Why is the project important?
- What’s the approach?

Project mission example

The Servers R Us project's mission is to design a new infrastructure for the company's server farm to make it scalable and flexible, and to implement the new design so that the server farm is operational by August 2020 to support the new development work planned for the fall.

Project Objectives

- **Specific**
- **Measurable**
- **Attainable/Acceptable (by stakeholders)**
- **Relevant/Realistic**
- **Time-bound**

Planning

- **What are we going to achieve?**
 - Project Scope statement => scope creep, deliverables, success criteria
- **What approach are we going to take?**
 - Project strategy
- **What are we going to do?**
 - WBS
 - Admin processes (managing risk, controlling changes, communicating and managing quality)

Planning

- **When will the project start and finish?**

- Project schedule

- **Who will work on the project**

- Responsibility assignment matrix

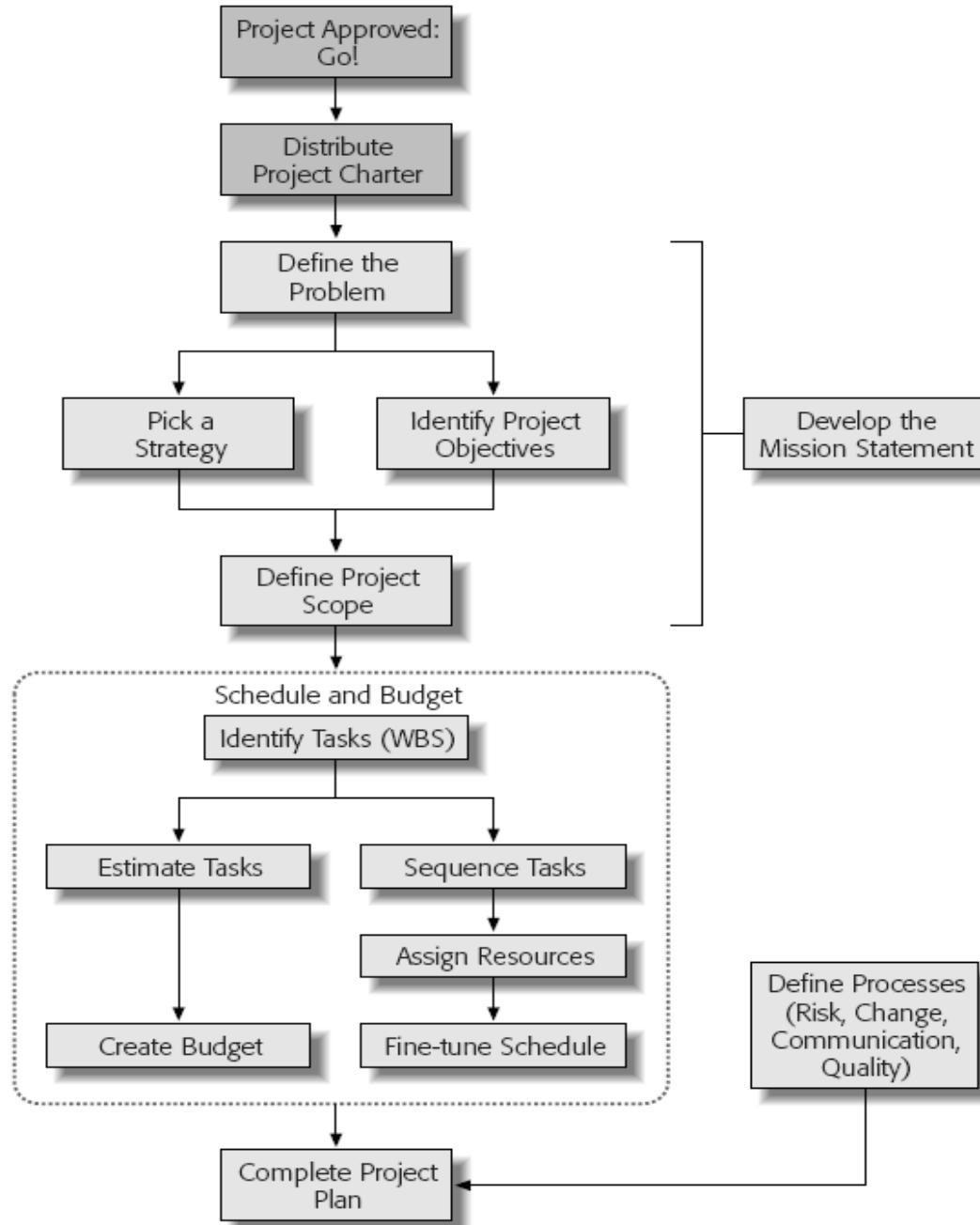
- Project organization chart

- **How much will it cost?**

- Budget

- **How good do the results need to be?**

- How you intend to achieve the level of quality the project requires, and how you'll measure that quality



Defining Project scope

■ Overall description of the work.

- Ex. This is where you state that the project is to “build a website.”

■ Deliverables.

- What will be produced by the project, and what are its key features? Also, what client need is the project satisfying?

■ Justification for the project.

- Sometimes it is necessary to dive into the justification of why the project was initiated in the first place.

Defining Project scope [2]

■ Constraints.

- If the project faces certain physical boundaries, these can be a source of risk and thus should be defined further.

■ Assumptions.

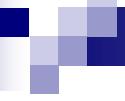
- All projects have assumed certain conditions as part of their existence. What are those assumptions and what impact does their inaccuracy have on the project?

■ Inclusions/Exclusions.

- Many projects have items that are uncertain. Projects sometimes do and sometimes don't include those things. Items need to be explicitly included or excluded from the project.

Examples

- Project Scope Statement Template
- Project Management book (Chapter 11 Project Charter)



Project strategy

- Is it feasible?
- Are the risks acceptable?
- Does the strategy fit the culture?

WBS

- Identifying the work to do
- Basis for
 - Laying out the project's schedule
 - Building a project team
 - Setting the project's budget

Communication Mgmt

- Who needs to know?
- What do they need to know?
- When do they need to know it?
- How should they receive it?

Communication plan

<i>What Information</i>	<i>Target Audience</i>	<i>When?</i>	<i>Method of Communication</i>	<i>Provider</i>
Milestone report	Senior management and project manager	Bimonthly	E-mail and hardcopy	Project office
Project status reports & agendas	Staff and customer	Weekly	E-mail and hardcopy	Project manager
Team status reports	Project manager and project office	Weekly	E-mail	Team recorder
Issues report	Staff and customer	Weekly	E-mail	Team recorder
Escalation reports	Staff and customer	When needed	Meeting and hardcopy	Project manager
Outsourcing performance	Staff and customer	Bimonthly	Meeting	Project manager
Accepted change requests	Project office, senior mgmt., customer, staff, and project mgr.	Anytime	E-mail and hardcopy	Design department
Oversight gate decisions	Senior management and project manager	As required	E-mail meeting report	Oversight group or project office



Change Management

- Submit change requests
- Receive and record change requests
- Evaluate the effects of change requests on cost, schedule, and quality
- Decide whether change requests become part of the project
- Update project documents to incorporate accepted changes
- Track changes as you do with other project tasks

Risk Management [1]

■ Identifying risks

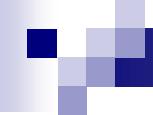
- Are there uncertainties in the plan? (“assume” makes an ass out of *u* and *me*)
- Are your choices limited?
- Are constraints significant?
- What level of experience do people have?
- Does the project depend on factors out of your control?

Risk Management [2]

- Assessing risks
 - Risk value = $f(\text{impact}, \text{probability})$
- Planning risk response
 - Accept
 - Avoid
 - Control (mitigate)
 - Transfer
 - Plan for alternatives (contingency)

Risk Management [3]

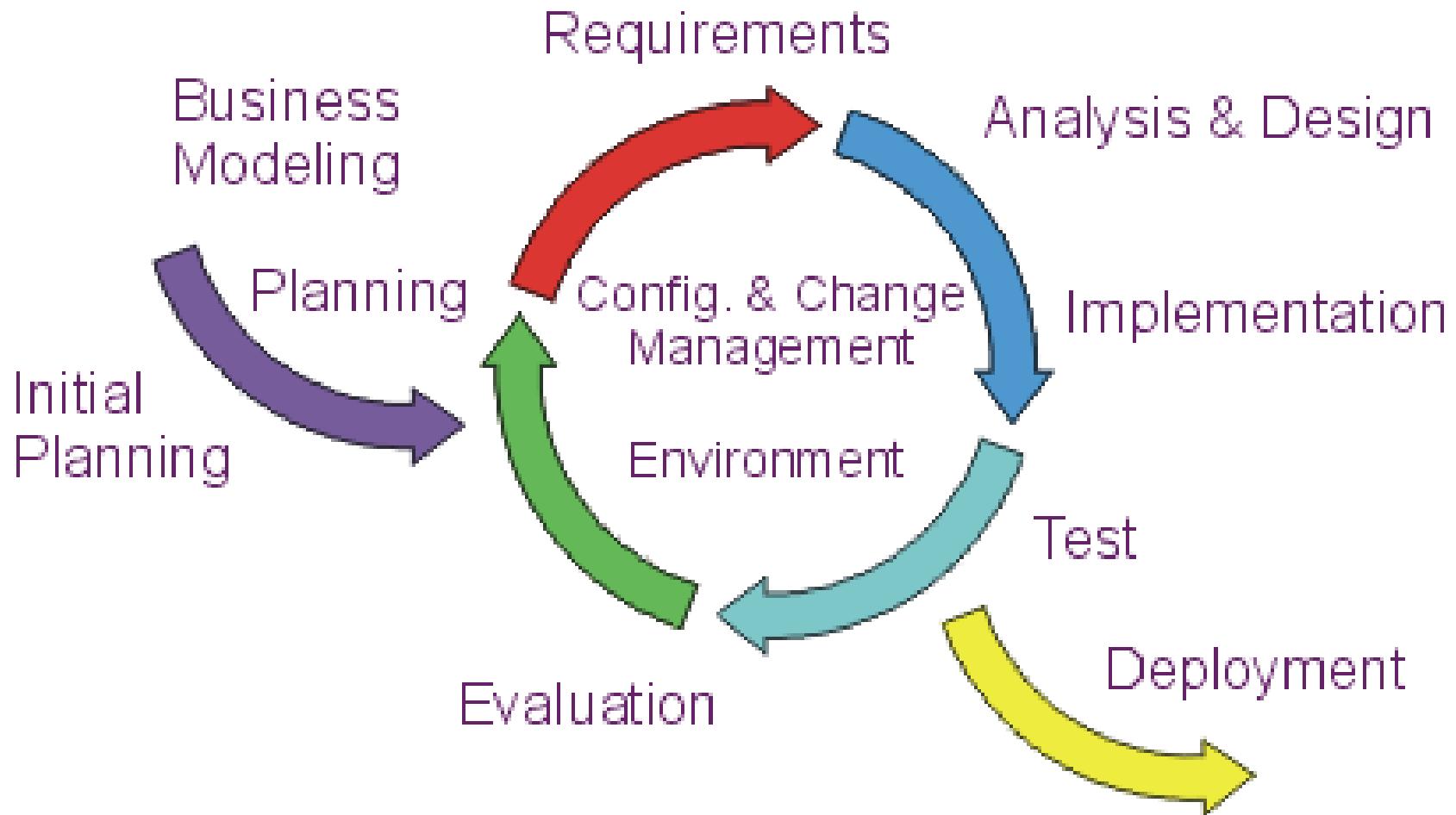
- Tracking risks
 - Risk log – risk, value, planned response, monitoring person, etc.



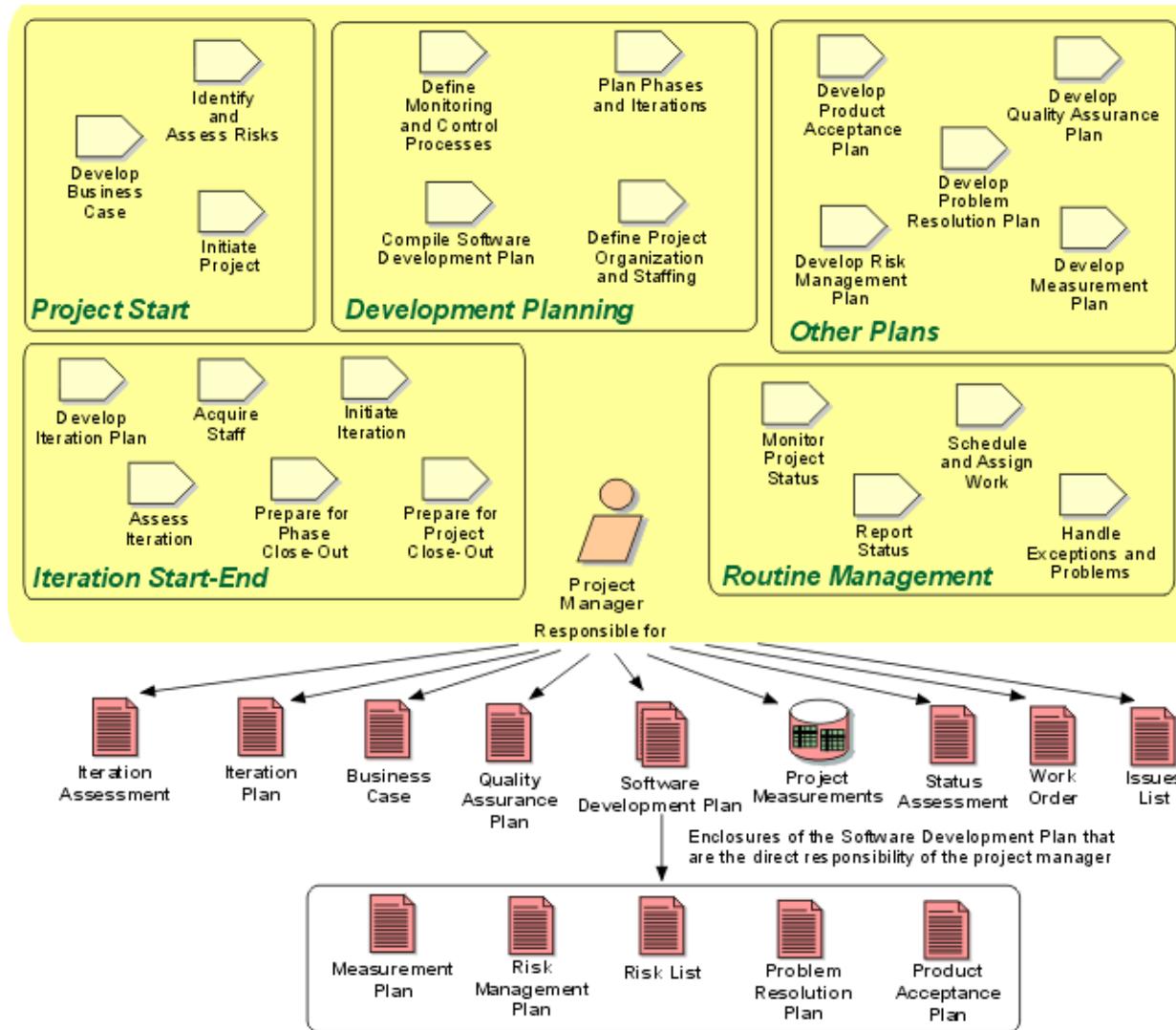
Let's see these translated in...

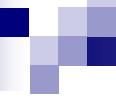
- RUP

RUP Development (© RUP)



A look at PM in RUP (© RUP)





Project Start::Develop Business Case

- Describe the Product
- Define the Business Context
- Define the Product Objectives
- Develop the Financial Forecast
- Describe the Project Constraints
- Describe Options

Project Start::Initiate Project

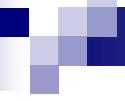
- Assign Project Review Authority (PRA)
- Assign project manager
- Assign project planning team (Project Manager, Software Architect, System Analysts, Development Lead, Test Lead, Configuration Management Manager, Customer representative)
- Approve product acceptance criteria

Project Start::Identify and Assess Risks

- Identify Potential Risks
- Analyse and Prioritize Risks
- Identify Risk Avoidance Strategies
- Identify Risk Mitigation Strategies
- Identify Risk Contingency Strategies
- Revisit Risks during the Iteration
- Revisit Risks at the End of an Iteration

Project Start::Develop Development Case

- Decide How to Perform Each Discipline
- Tailor Artefacts per Discipline
- Modify Disciplines and Activities
- Choose Lifecycle Model
- Identify Stakeholders
- Map Roles to Job Positions
- Describe Sample Iterations
- Document the Development Case
- Maintain the Development Case

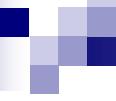


Development Planning::Define Monitoring and Control

- Define project "indicators"
- Define sources for project indicators
- Define procedure for team status reporting
- Define procedure & thresholds for corrective action
- Define procedure for project status reporting

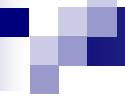
Development Planning::Plan Phases and Iterations

- Estimate Project
- Define Project Phase Milestones
- Define Milestone Goals
- Define Number, Length, and Objectives of Iterations Within Phases
- Refine Milestones Dates and Scope
- Determine Project Resourcing Requirements
- Develop Project Close-Out Plan



Development Planning::Define Project Organization and Staffing

- Define Project Organization
- Define Staffing Requirements



Development Planning::Compile Software Development Plan

- Develop SDP project management content
- Develop enclosed project management plans
- Coordinate development of supporting plans

Other Plans::Develop Product Acceptance Plans

- Define customer and project responsibilities
- Document the product acceptance criteria
- Identify artifacts and evaluation methods
- Identify Required Resources
- Define product acceptance schedule
- Define problem resolution process

Other Plans::Develop Risk Management Plan

- Define risk management procedure & tools
- Create initial risk list
- Assign risk management team
- Decide strategies for managing top 10 risks
- Define risk indicators for top 10 risks
- Set schedule for risk reporting and reviews

Other Plans::Develop Measurement Plan

- Define the Primary Management Goals
- Validate the Goals
- Define the Subgoals
- Identify the Metrics Required to Satisfy the Subgoals
- Identify the Primitive Metrics Needed to Compute the Metrics
- Write the Measurement Plan
- Evaluate the Measurement Plan
- Put in Place the Collection Mechanisms



Other Plans::Develop Problem Resolution Plan

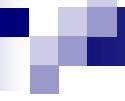
- Define Problem Resolution Procedure(s)
- Select Tracking Tools and Techniques
- Assign Problem Management Team(s)
- Set Schedule for Problem Management Activities

Other Plans::Develop Quality Assurance Plan

- Ensure Quality Objectives are Defined for the Project
- Define Quality Assurance Roles and Responsibilities
- Coordinate with Developers of Referenced Plans
- Define Quality Assurance Tasks and Schedule

Iteration Start End::Develop Iteration Plan

- Determine the Iteration Scope
- Define Iteration Evaluation Criteria
- Define Iteration Activities
- Assign Responsibilities



Iteration Start End::Acquire Staff

- Staff the Project
- Map Staff Skills to Roles
- Form Teams
- Train Project Staff

Iteration Start End::Initiate Iteration

- Assign staff to work packages
- Acquire and assign non-personnel resources
- Issue work orders

Iteration Start End::Assess Iteration

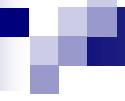
- Collect Metrics
- Assess the Results of the Iteration
- Consider External Change
- Examine the Evaluation Criteria
- Create Change Requests

Iteration Start End::Prepare for Phase Closeout

- Check Status of Required Artifacts
- Schedule Configuration Audit
- Conduct a Phase Post-Mortem Review
- Distribute Artifacts to Stakeholders
- Complete Lifecycle Milestone Review Action Items

Iteration Start End::Prepare for Project Closeout

- Update Project Close-Out Plan and Schedule Activities
- Schedule Final Configuration Audit
- Conduct a Project Post-Mortem Review
- Complete Acceptance Action Items
- Close Out the Project

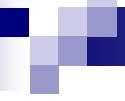


Routine Management::Monitor Project Status

- Capture work status
- Derive progress indicators
- Derive quality indicators
- Evaluate indicators vs. plans

Routine Management::Schedule and Assign Work

- Allocate Change Request to an Iteration
- Assign Responsibility
- Describe Work and Expected Outputs
- Budget Effort and other Resources
- Set Schedule
- Re-plan
- Issue Work Order



Routine Management::Report Status

- Prepare Status Assessment

Routine Management::Handle Exceptions and Problems

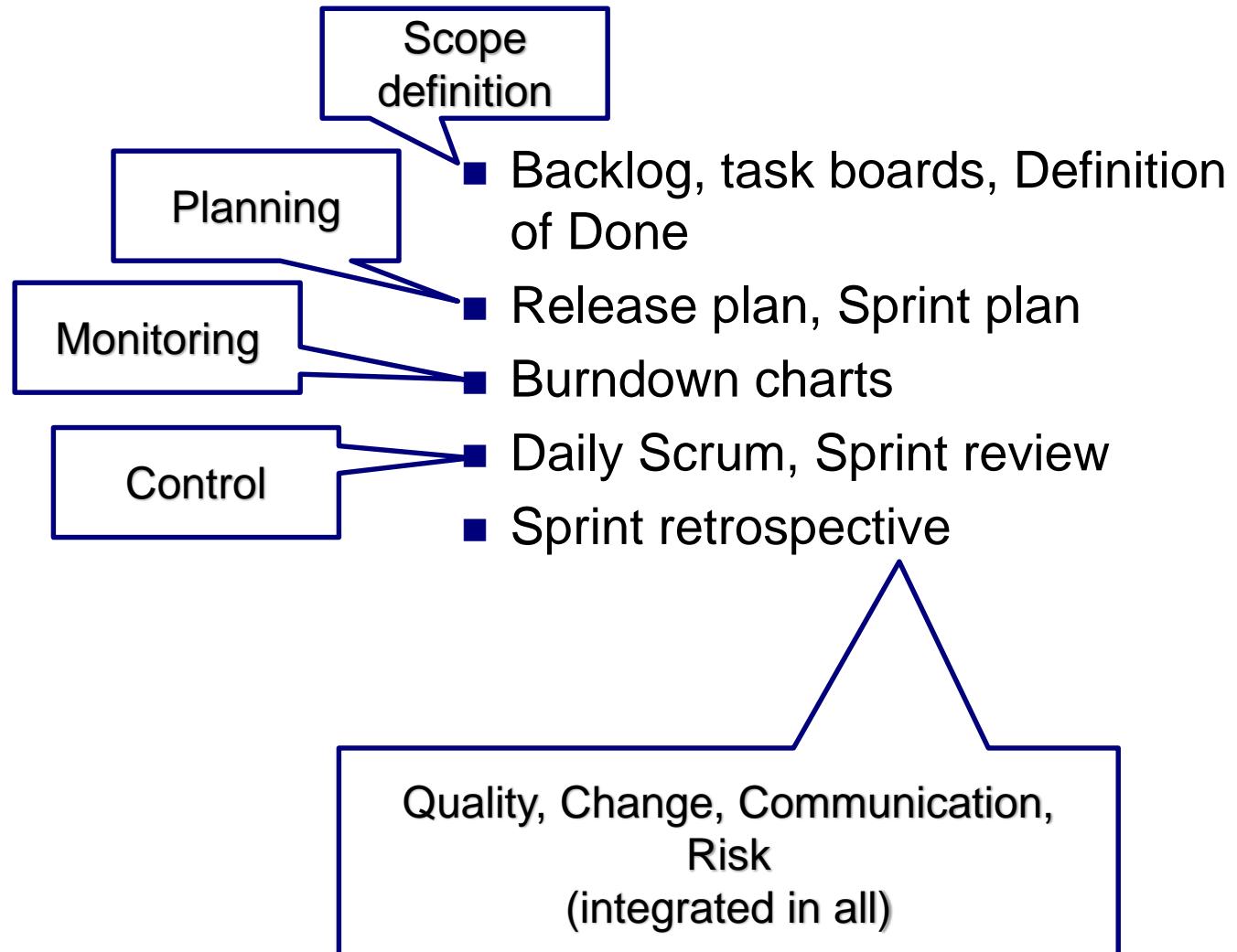
- Evaluate exceptions and problems
- Determine appropriate corrective actions
- Issue Change Requests and/or Work Orders

Agile

- Are the PMBOK processes applicable to Agile methodologies?

SCRUM

- Scope
- Integration
- Time
- Cost
- Quality
- Change
- Communication
- Risk



Problem type 1

- Adopt Agile for the **wrong reason** and without necessary consideration for local context.
- Choose Agile practices to implement **by convenience**, thus lead to incomplete and broken implementation.
- **Lack of organizational level** management on Agile adoption.
- **Lack of necessary discipline**, too often try go fast by cutting corners.

Need to Adopt the Whole not Part of Agile

One can not develop a releasable slice of software with value in 1 to 4 weeks without doing all of the following:

- An **effective cross functional** team who knows its capability (velocity).
- **Expose issues and problems** via daily meetings and resolve them within the same day if possible.
- **Refine user stories** so that the top ones are small enough to be done within days.
- Establish a **clearly defined “Done” criteria** to ensure quality and control of technical debt.
- Make **Continuous Integration** routine to get real time feedback and quick deployment.
- **Block interruption** to team members so they can focus on achieving sprint goal.

Problem type 2: weakness in the method!

- **Underestimate the tasks** in more complex setting
(For example, Scrum (to a degree) trivializes product planning and requirement analysis with the role of PO and Product Backlog dealing with multiple systems or products).
- **Lack of workflow** (in Scrum) may not be able to expose bottlenecks in real time.
- **Lack of organizational level improvement management** and org. level **retrospective reviews**

CMMI as support for Agile

- Problem (Planning)

- Project estimates are unrealistic or unknown

- Guideline

- Extract the high-priority user stories from the product Backlog
 - Use Team Estimating Game/Planning poker to estimate the user-stories
 - Assign the appropriate number of user story points in Release/Sprint planning (based on the team velocity)

CMMI as support for Agile

■ Problem (Control)

- Project are not delivered in schedule

■ Guideline

- Monitoring on different granularity levels
- Daily stand up meetings => review burn down charts
- Sprint review => micro-grooming for smaller and more detailed user stories
- Sprint retrospective => backlog grooming to break down user stories for the next sprints

CMMI as support for Agile

- Problem (Requirements management)
 - Requirements are changing frequently
- Guideline
 - Product Backlog contains all user stories
 - All changes are integrated in the Backlog by the PO
 - During Sprint Planning the user stories are selected and included in the Sprint Backlog.
 - After Planning, the Sprint Backlog is frozen. Any additional change is included in the Product Backlog
 - The Product Backlog is groomed regularly

CMMI as support for Agile

- Problem(Quality management)
 - Customers are not satisfied with the end product
- Guideline
 - Clear/verifiable acceptance criteria for all user stories
 - During Sprint Demo/Reviews all user stories should be validated
 - If customer is still not happy, new user stories should be added to the Backlog

CMMI as support for Agile

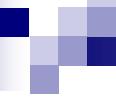
- Problem (Resource management)
 - Unable to secure or retain project resources
- Guideline
 - During Release Planning ensure the necessary resources (team members with needed skills)
 - New issues should be raised in Daily stand ups

CMMI as support for Agile

- Problem (Communication management)
 - Poor project communication
- Guideline
 - Agile methods focus on real-time communication
 - Several techniques: Release/Sprint Planning, Daily Stand up, Spring Demo/Reviews, Retrospectives
 - Any additional communication related decisions can be added in the Team Agreement

CMMI as support for Agile

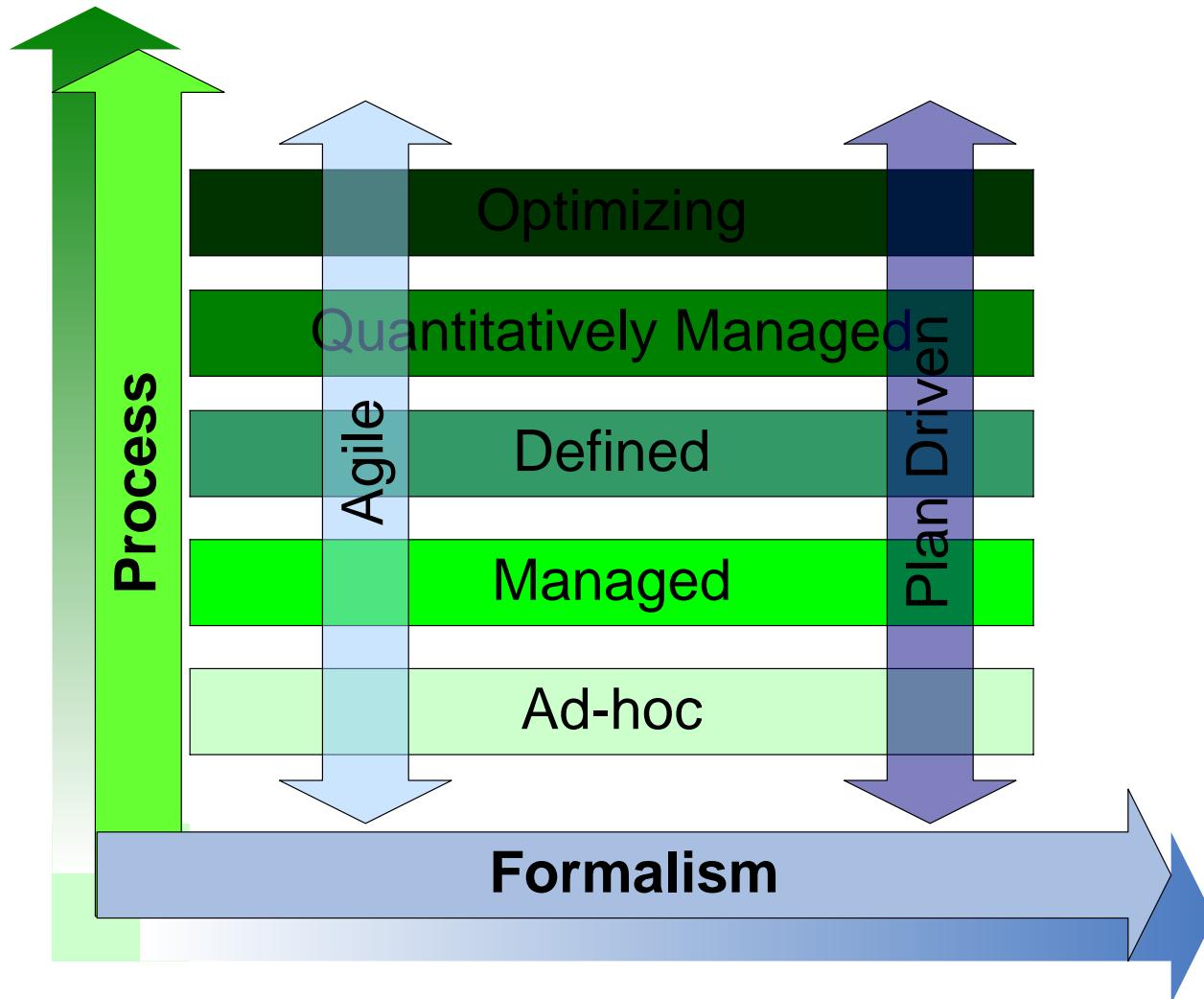
- Problem (Risk management)
 - Failure to foresee potential problems
- Guideline
 - Raise issues in daily scrum meetings
 - Scrum Masters should elicit and address raised issues
 - Latent issues (i.e. technical debt) should be added to the Product Backlog as story (or other work item)



Projects are unique

- What is the right process?

Tailor the process & methodologies

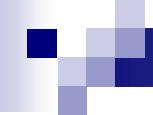


Tailoring the process

- **Size.** The number of control elements in the methodology. Each deliverable, standard, activity, quality measure, and technique description is an element of control.
- **Ceremony.** The amount of precision and the tightness of tolerance in the methodology. Greater ceremony corresponds to tighter controls.
- **Weight.** The conceptual product of size and ceremony.
- **Precision.** How much you care to say about a topic?
- **Accuracy.** How correct you are when you speak about a topic?
- **Relevance.** Whether or not speak about a topic?
- **Tolerance.** How much variation is permitted in the execution of the methodology?

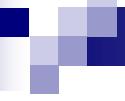
References

- *Project management : the managerial process* by Clifford F Gray; Erik W Larson, 5th edition
- *Software Project Management: A Unified Framework*, Walker Royce, Addison Wesley
- *Agile Project Management with SCRUM*, Ken Schwaber, Microsoft Press
- *A Guide to The Project Management Body of Knowledge*
- *CMMI®: Guidelines for Process Integration and Product Improvement*
- *Agile Software Development*. Cockburn, Alistair. 2002.
- A Guide to Scrum and CMMI®: Improving Agile Performance with CMMI, 2016



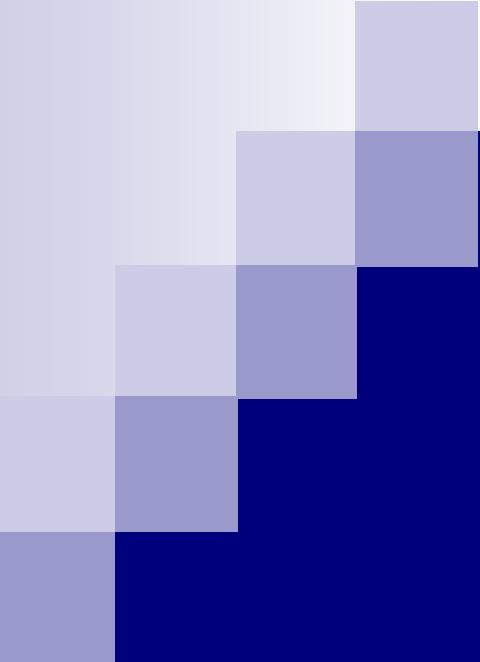
Quiz time

- Please switch over Moodle



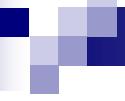
Discussion topic next week

- Consider your diploma project. Provide a project Charter. Show how the project objectives are SMART.



Planning and tailoring the process

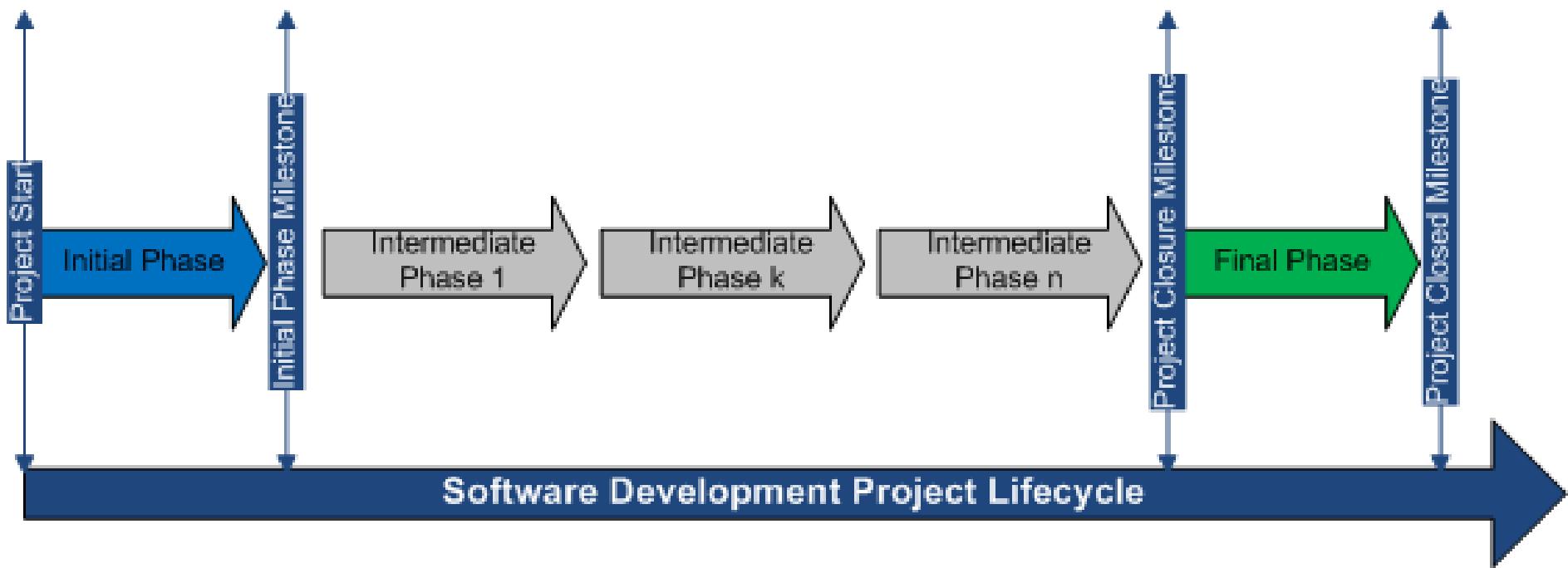
Lecture 4



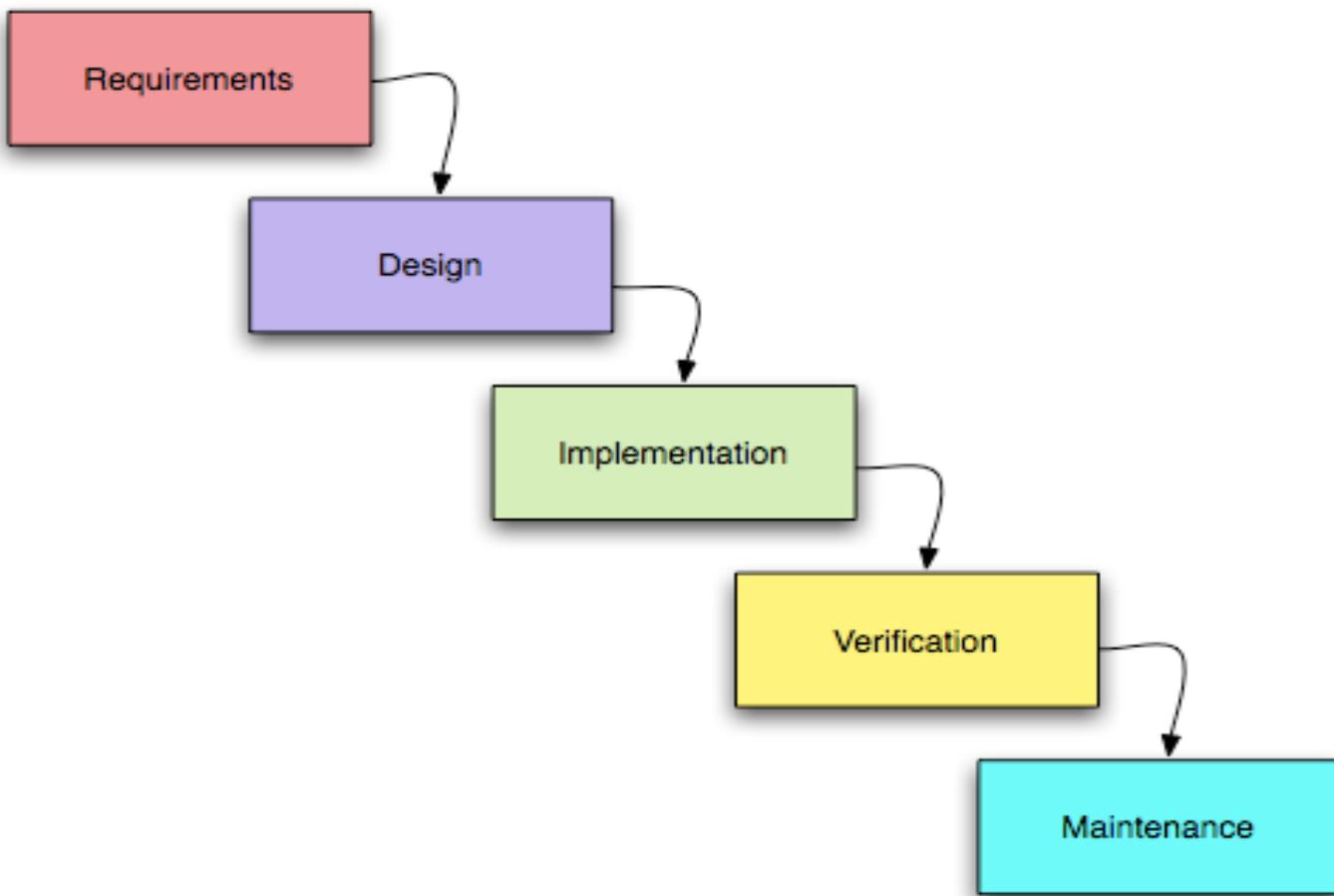
Recap

- First some recap
- ... in pictures

Recap: Project Lifecycle



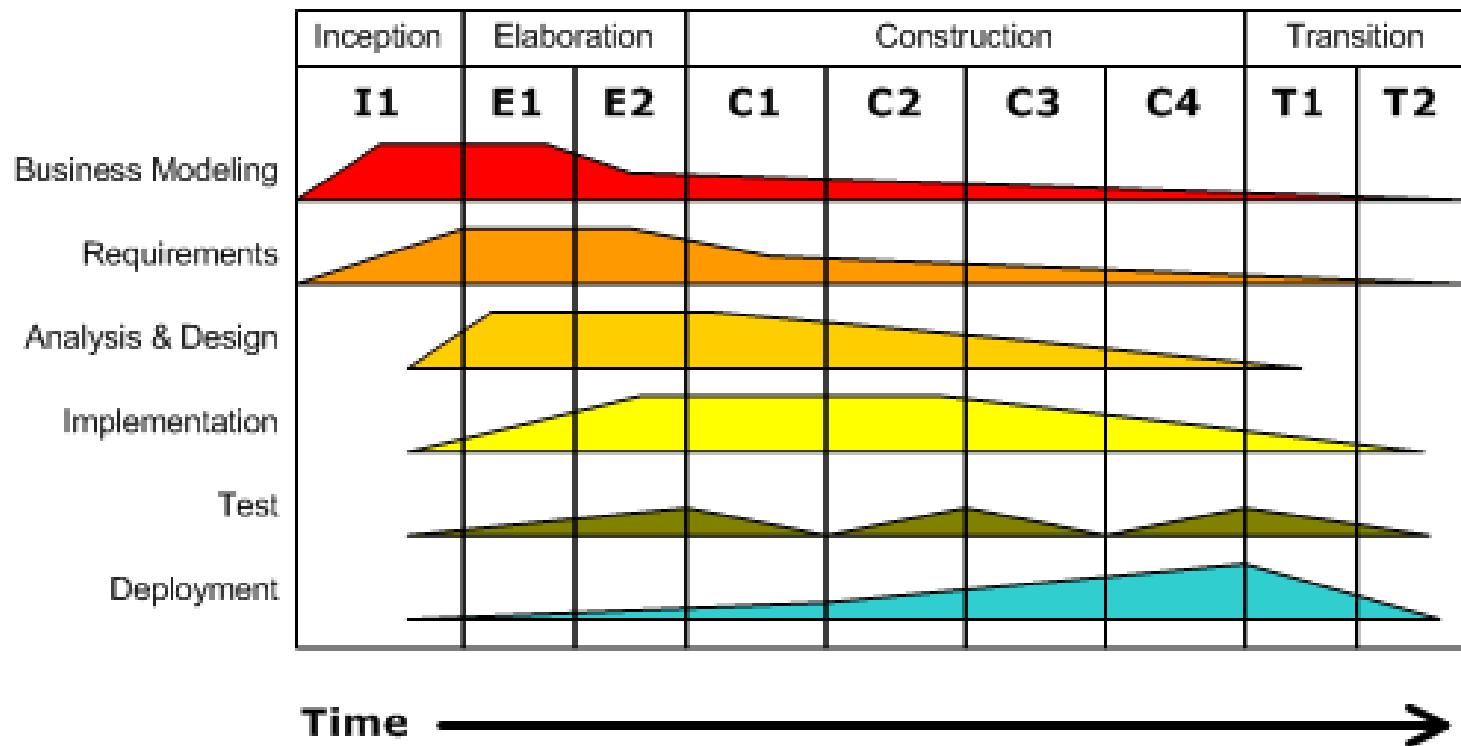
Recap: Waterfall Model



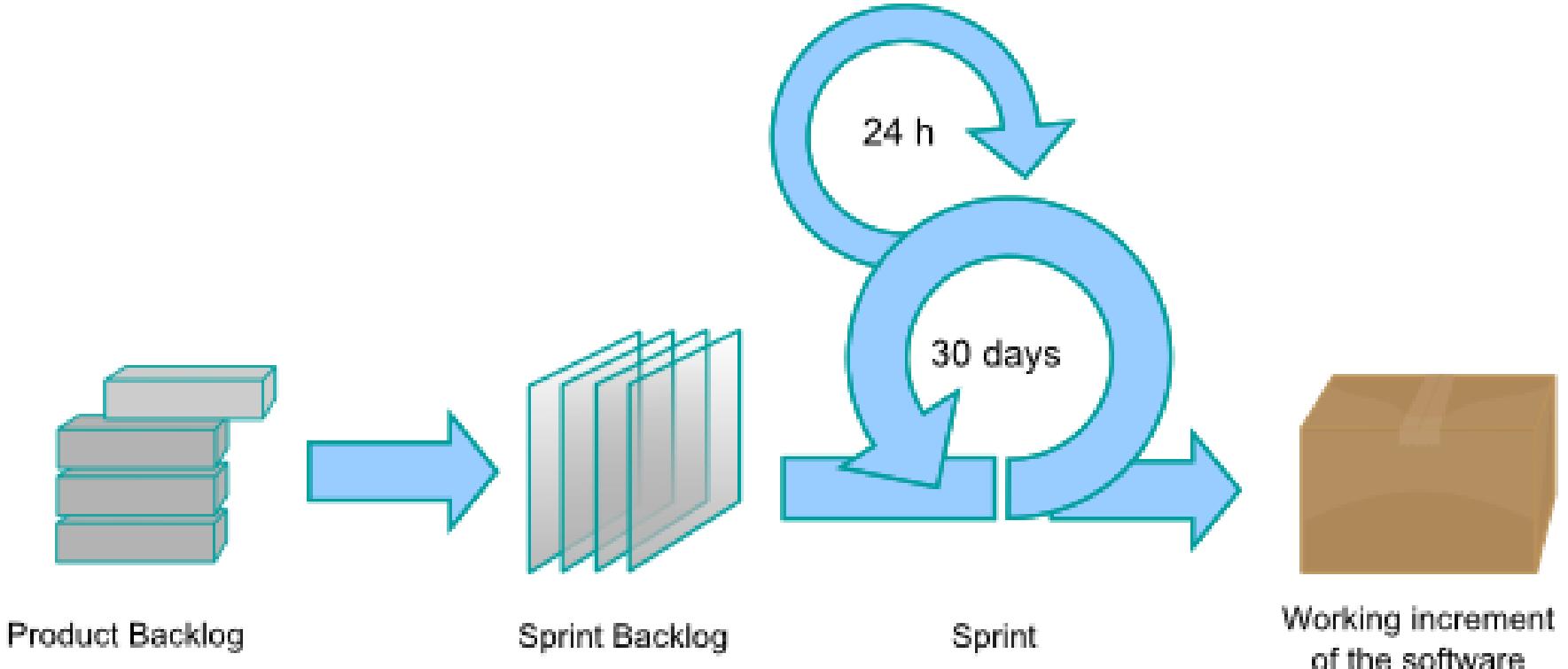
Recap: UP – Iterative, incremental

Iterative Development

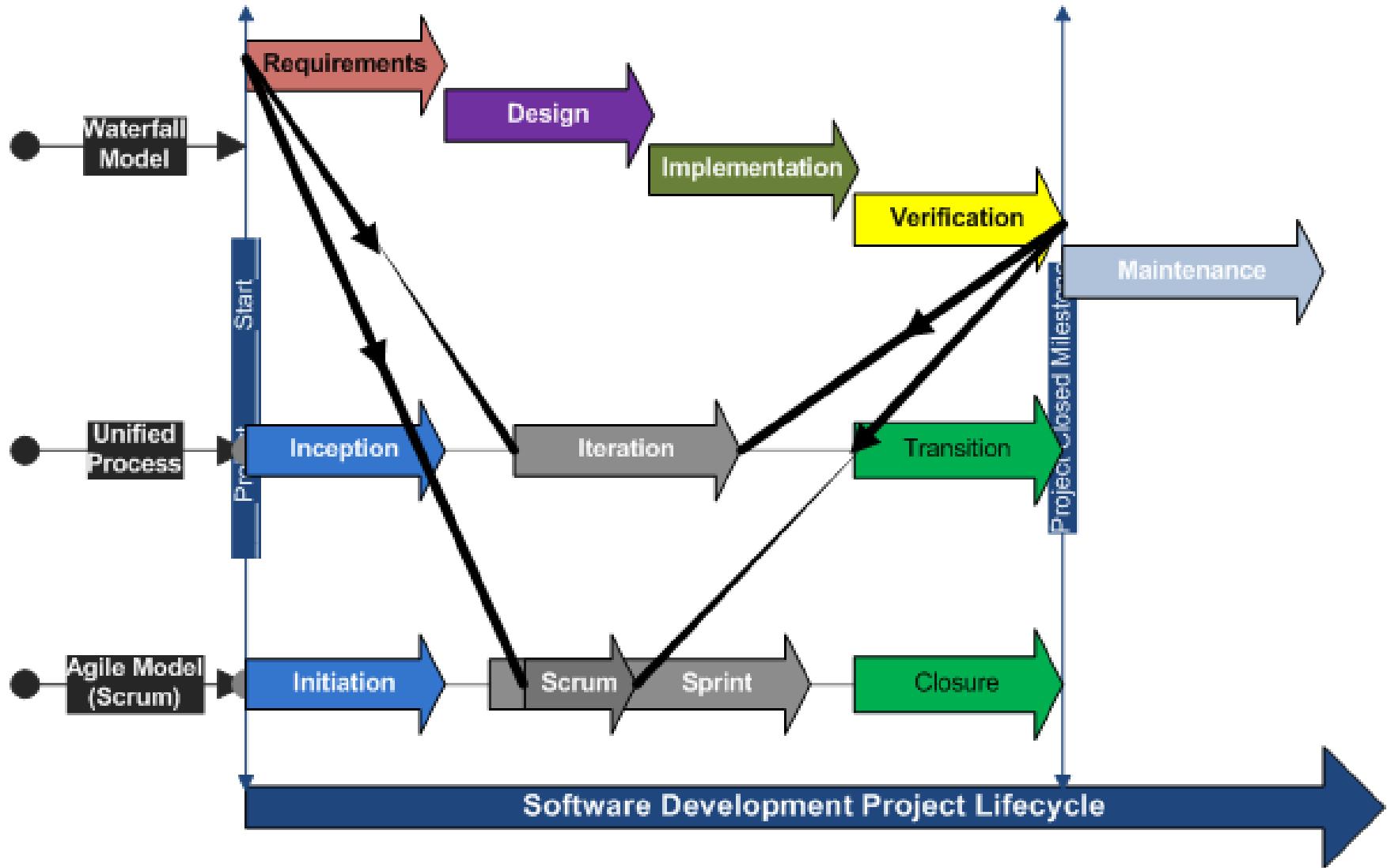
Business value is delivered incrementally in time-boxed cross-discipline iterations.



Recap: Agile model - SCRUM



Waterfall – UP - Agile



Tailoring guidelines

- **Size.** The number of control elements in the methodology. Each deliverable, standard, activity, quality measure, and technique description is an element of control.
- **Ceremony.** The amount of precision and the tightness of tolerance in the methodology. Greater ceremony corresponds to tighter controls.
- **Weight.** The conceptual product of size and ceremony.
- **Precision.** How much you care to say about a topic?
- **Accuracy.** How correct you are when you speak about a topic?
- **Relevance.** Whether or not you speak about a topic?
- **Tolerance.** How much variation is permitted in the execution of the methodology?

Today's Topics

- Planning and Tailoring the Process
 - Goals, Roles and Activities
- Examples
 - Development Case
 - Disciplined Agile Delivery

References

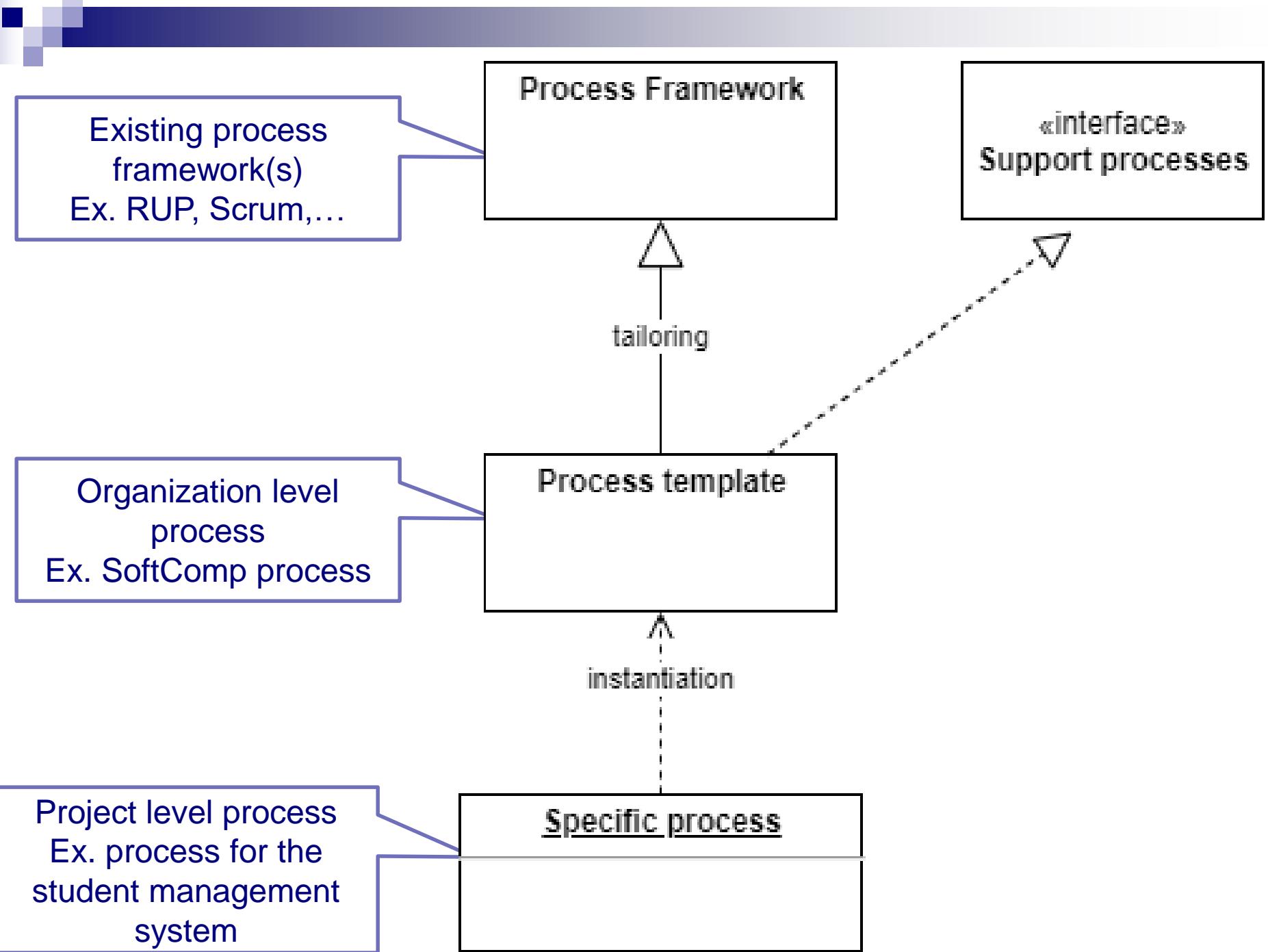
- PMBOK, RUP
- IQuest white papers
- Software Process Tailoring: Towards a Model-based Approach, P.Bannerman, R.Jeffery, 4th ISCAS/USC/UMAS/NICTA Joint Workshop, Beijing, 2012 [SPTTMA]
- Project and Process Tailoring For Success, Chicago Quality Assurance Association [CQAA]
- Kalus, Georg, and Marco Kuhrmann. "Criteria for software process tailoring: a systematic review." *Proceedings of the 2013 International Conference on Software and System Process*. ACM, 2013.
- <https://www.pmi.org/disciplined-agile/process/introduction-to-dad>

Initial approach

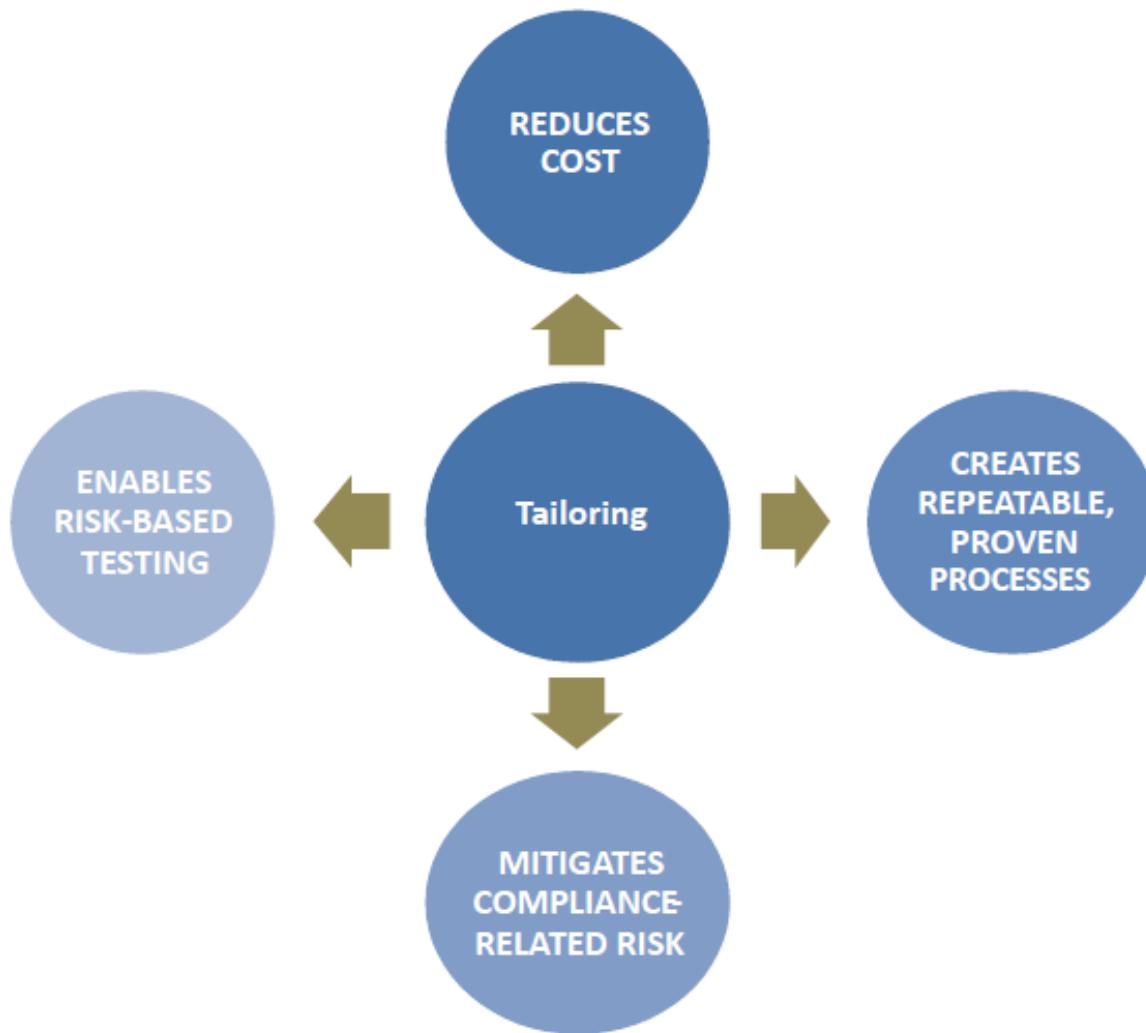
- **Most companies start out with the good intention of creating and standardizing on a single process:**
 - This quest for uniformity and economies-of-scale quickly backfires
 - Always need to create the same 20+ page requirements document
 - Always needs to create the same 20+ page test plan
 - Organizations are buried in process related documents, checklists, templates
 - Everything slows to a sluggish pace

Process tailoring vs. instantiation

- Tailoring
 - Defining a **template process at the organizational level** starting from existing process frameworks (ex. RUP, etc.)
- Instantiation
 - Defining the **specific process for a given project** starting from the organizational template process.



Tailoring (instantiating) [CQAA]



Reduces Cost

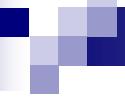
- By aligning process intensity with project risk and complexity, tailoring can reduce demands for: Forms, Checklists, Processes, Procedures, Templates
- Can free valuable time for engineering and testing resources that can cause small projects to proceed at a sluggish pace.
- On average, tailoring can reduce process intensity by 3X-6X which can equate to more than 20% savings in project costs and other costs associated with standards, compliance and project oversight.

Mitigate compliance related risk

- Pre-populating schedules with **compliance-related processes, templates and policies** based on knowledge of the compliance/standards teams, not relying solely on the project lead
- Enabling the project team to conduct **pre-audit run-throughs** where gaps in compliance can be highlighted and addressed prior to critical and visible audits or checkpoint/Authorization-to-Proceed (ATP) meetings
- Providing a **Compliance Checklist** that enables the team to clearly gauge progress towards compliance using a Red/Yellow/Green model

Why tailoring/instantiating a process?

- Consistent project management is always about following a process
- Company processes are applicable for any of the project types developed by the company
- Applying the FULL company process is most of the times counterproductive
- If the process slows you down you got the tailoring/instantiation wrong



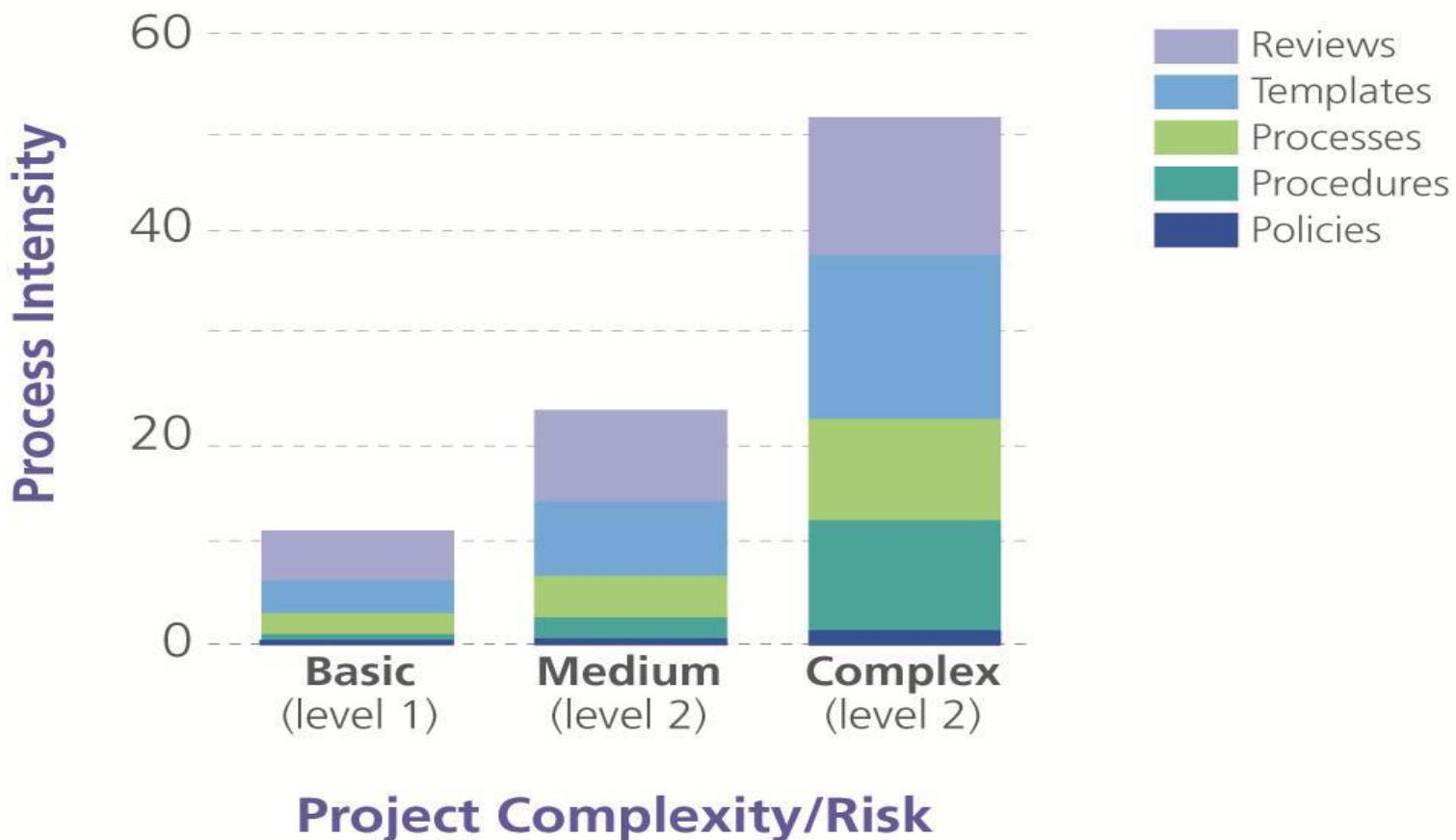
Process Tailoring Overview

- ... PMBOK

Process Driven

- You have to tailor every process described in the PMBOK
- Integration, Scope, Time, Cost, Quality, Risk, Human Resources, Communication, Procurement

Intensity vs. Complexity



How do we approach tailoring?

- We need a systematic approach (a process)
- Project Management Scalable Methodology Guide
- For every process area a specific priority is assigned based on the project specifics
- Example:

For Scope Management we should tailor:
Requirements Definition, WBS, Product and
Project Control

Example – Tailoring Scope Management

Area	Priority			
	4 <i>Minor investment, requirements are well understood and reasonably stable; familiar internal customer.</i>	3 <i>Moderate investment, requirements not obvious; potential for misunderstandings; familiar customer.</i>	2 <i>Significant investment, significant requirements; probable changes; significant technical unknowns</i>	1 <i>Major investment, extensive requirements; volatile environment; substantial visibility and technical risks.</i>
Requirements definition	<ul style="list-style-type: none"> • Document the requirements definition; • obtain approval from key stakeholders; • place requirements document under version identification and change control. 	<ul style="list-style-type: none"> • Document business case and detailed performance based requirements specifications; • use modular structure for documenting functions, performance and features. 	<ul style="list-style-type: none"> • Review detailed requirements definition with customer and sponsor; • employ walk-thrus, simulations, prototypes, demonstrations, mock-ups or draft user documentation 	<ul style="list-style-type: none"> • Conduct a rigorous, extensive requirements definition and review process; • map requirements to design and test documents; • conduct independent peer reviews and formal customer approval.

Example – Tailoring Scope Management

Area	Priority			
	4 <i>Minor investment, requirements are well understood and reasonably stable; familiar internal customer.</i>	3 <i>Moderate investment, requirements not obvious; potential for misunderstandings; familiar customer.</i>	2 <i>Significant investment, significant requirements; probable changes; significant technical unknowns</i>	1 <i>Major investment, extensive requirements; volatile environment; substantial visibility and technical risks.</i>
Work Breakdown Structure	<ul style="list-style-type: none">• Prepare WBS to level 3 to ensure comprehensive identification of tasks and outputs;• firm-up WBS structure early.	<ul style="list-style-type: none">• Use WBS to prepare responsibility matrix, cost estimates, and schedules;• publish WBS as a project baseline document.	<ul style="list-style-type: none">• Prepare WBS dictionary to level 3;• use WBS structure to aggregate cost data.	<ul style="list-style-type: none">• Use <i>product-oriented</i> WBS to organize requirements, schedules, budgets, testing, and deliverables;• map WBS to organizational breakdown structure.

Example – Tailoring Scope Management

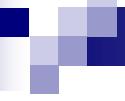
Area	Priority			
	4 <i>Minor investment, requirements are well understood and reasonably stable; familiar internal customer.</i>	3 <i>Moderate investment, requirements not obvious; potential for misunderstandings; familiar customer.</i>	2 <i>Significant investment, significant requirements; probable changes; significant technical unknowns</i>	1 <i>Major investment, extensive requirements; volatile environment; substantial visibility and technical risks.</i>
Product Baseline Control	Use requirements document to establish baseline stability; have PM approves major changes; establish version control as product design matures.	Place requirements document under formal control; require change request approval by PM; report metrics to track scope changes.	Establish configuration identification, status accounting, control process , and configuration audits; staff a formal CM function.	Establish baseline for requirements, functional and allocated specs, and product design; evaluate ECP impact and require functional approvals prior to CCB; manage the pace of changes.

Example – Tailoring Scope Management

Area	Priority			
	4 <i>Minor investment, requirements are well understood and reasonably stable; familiar internal customer.</i>	3 <i>Moderate investment, requirements not obvious; potential for misunderstandings; familiar customer.</i>	2 <i>Significant investment, significant requirements; probable changes; significant technical unknowns</i>	1 <i>Major investment, extensive requirements; volatile environment; substantial visibility and technical risks.</i>
Project Baseline Control	Maintain historical track of cost and schedule estimate revisions; report baseline and latest revised estimates against actual.	Identify cost and schedule baselines; report metrics to show changes against milestone estimates.	Establish cost and schedule baselines and maintain disciplined controls; report all baseline re-plans or changes; define tasks in discrete work packages.	Establish firm cost and schedule baseline between major milestones; require sponsor signature for baseline re-plans; use work package approval and authorization process.

Assessment

- After 6 –12 months recalculate the intensity ratio
 - Ideally 5.0 or higher indicating a differentiation in effort between simplest and more complex projects
- Based on the analysis the organizations should:
 - Identify new assets to be created
 - Modify existing assets to better meet needs
 - Improve training for all involved in the process
 - Explore automation to reduce manual effort



Process Tailoring

- ... RUP

Roles

■ Process Engineer

- Defines the template process of the organization
- Tailors the template process to match the specific needs of the project.
- Educates and mentors project members on process related issues.
- Ensures that valuable project experience is harvested and fed back into the template process.
- Assists the Project Manager in planning the project

Roles

■ Project Manager

- ... on small projects takes the role of process engineer
- Validates that the tailored process fits the project
- Fine tunes the tailoring
- Provides feedback to the process engineer to update the template process

Process Activities

- The following process activities occur in a project:
 - Tailor Instantiate the process for the project
 - Develop the development case (tailored instantiated methodology)
 - Prepare templates for the project
 - Prepare guidelines
 - Launch the development process



Artifacts

- Development Organization Assessment
(cross - project)
- **Development Case**
- Project Templates
- Project Guidelines
- Feedback to template process (improve
the organization process)

Tailor Instantiate the process

■ Goals

- To right-size the software development process according to the specific needs of the project
- To provide a relevant and accessible process description for the members of the project

Tailor Instantiate the process

■ Activities

- Analyze the Project
- Define the Scope of the Process
- Extend the Process Framework (optional)
- Configure the Process
- Prepare the Process for the Project
- Introduce the Process to the Project Members
- Maintain the Process

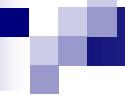
Develop the Development Case

■ Goals

- To develop a development case that describes the software-development process for a project (or projects).
- To relate the development case to the organization-specific process.

■ Activities

- Decide How to Perform Each Discipline
- Tailor Artifacts per Discipline
- Modify Disciplines and Activities
- Choose Lifecycle Model
- Identify Stakeholders



Develop the Development Case

■ Activities (continued)

- Map Roles to Job Positions
- Describe Sample Iterations
- Document the Development Case
- Maintain the Development Case

Prepare templates for the project

■ Goals

- harvest existing or develop new templates for use by the project.
- prepare the templates for project use by partially instantiating them with project-specific information.
- make the existing templates accessible to the project members when needed.

■ Activities

- Identify Templates for the Project
- Prepare Templates for Project Use
- Maintain Templates

Prepare guidelines for the project

■ Goals

- harvest existing or develop new guidelines for use by the project.
- make the existing guidelines accessible for the project members when needed.

■ Activities

- Identify the Project's Needs for Guidelines
- Prepare Guidelines for Project Use
- Maintain Guidelines

Launch the development process

■ Goal

- make the project members use the development process tailored for the project, together with the supporting tools.

■ Activities

- Make the changes public
- Educate project members
- Collect feedback

Guidelines discussion (1)

- Business Modeling Guidelines
 - Describes how you should model business use cases, business workers, and business entities.
 - Should be considered when the project needs to formally model the business to build a new system. The degree of business process redesign, or the complexity of the business process, dictates how comprehensive they need to be.

Guidelines discussion (2)

- Use-Case Modeling Guidelines
 - Needed whenever use cases will play a significant part in capturing the behavior of the system.
 - Should contain modeling conventions such as relationships to use, styles to follow for textual descriptions.

Guidelines discussion (3)

■ Design Guidelines

- A product of the architecture definition. It describes the guidelines to be followed during design, architectural design, and implementation.

Guidelines discussion (4)

■ Programming Guidelines

- Specific to the actual implementation language(s) and class libraries selected for the project.
- should specify
 - how to present code layout and commenting,
 - how to use naming conventions,
 - how to use language features.
 - precautions regarding certain language features.

Guidelines discussion (5)

- User-Interface Guidelines
 - Should give project-specific rules and recommendations for building the user interface.
- Often reference external publications (ex. Microsoft Surface User Experience Guidelines, Windows Mobile User Interface Guidelines, Windows Phone User Experience Design Guidelines, Windows Touch User Interface Guidelines, Windows User Experience & User Interface Guidelines)

Guidelines discussion (6)

■ Tool Guidelines

- Describe how the project makes the best use of the selected tool set. Will often include:
 - Installation information, such as version, configuration parameters,
 - Limitations in functionality, and functionality that the project decided not to use
 - Workarounds
 - Integration with other tools including procedures to follow, software to use, and principles to apply.

Guidelines discussion (7)

■ Test Guidelines

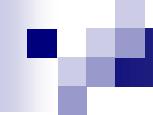
- Used to record adjustments (often tactical) to the way the test process is enacted on a given project, and to capture project-specific practices discovered during the dynamic enactment of the test process.
- Examples of test guidelines are test completion criteria and defect management guidelines.

Development Case

- The purpose of the Development Case is to capture the tailored process for the individual project.
=> you modify the process and adapt the terminology.
- The Development Case is created early in the Inception phase and is updated throughout the project as needed

Building a Development Case

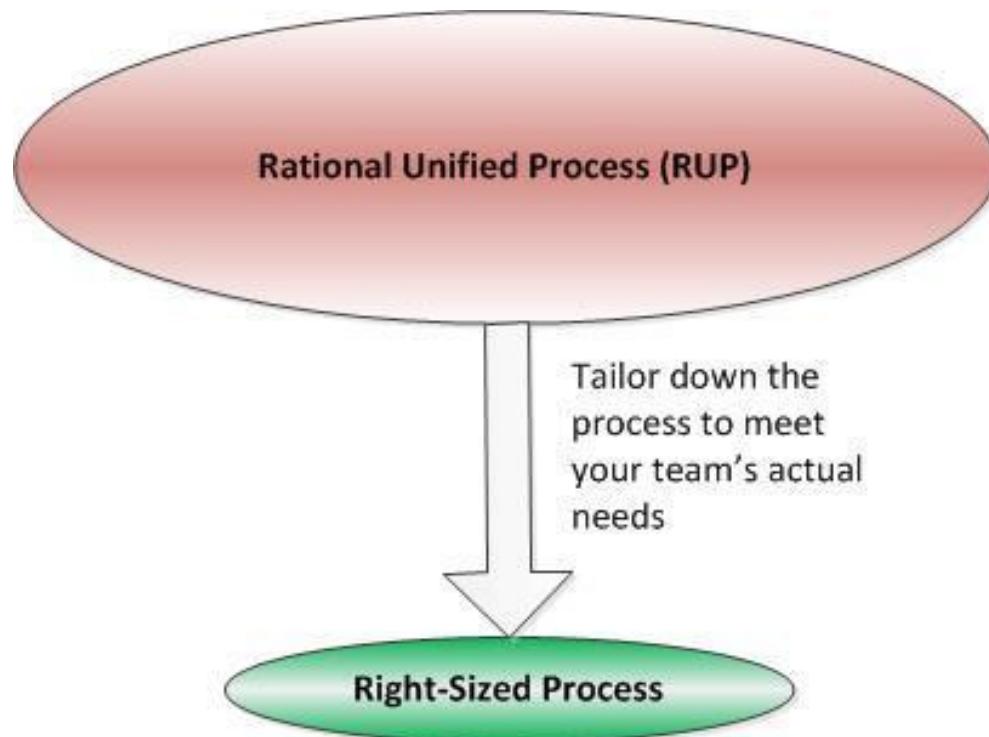
- The development case should not capture the entire process
- Responsibility and decisions about the process artefacts are delegated to members of the development team
- One reason for having a project process description at all is so several people can share information
- If process does not live in projects then the cost of maintaining the process description may be too high



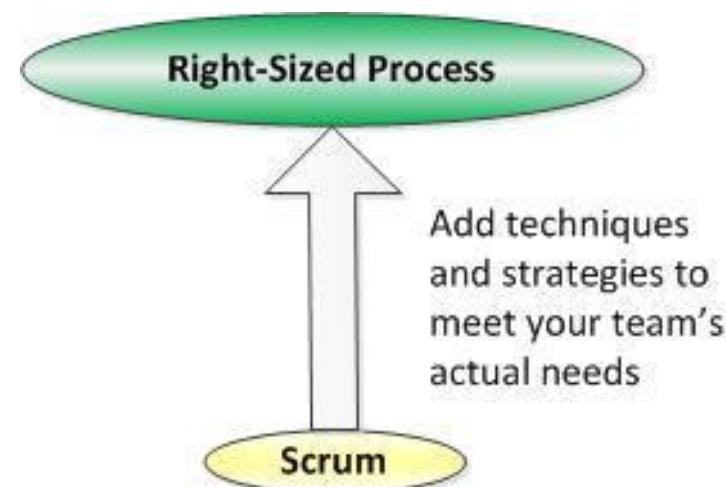
Development Case Examples

- Development Case.doc

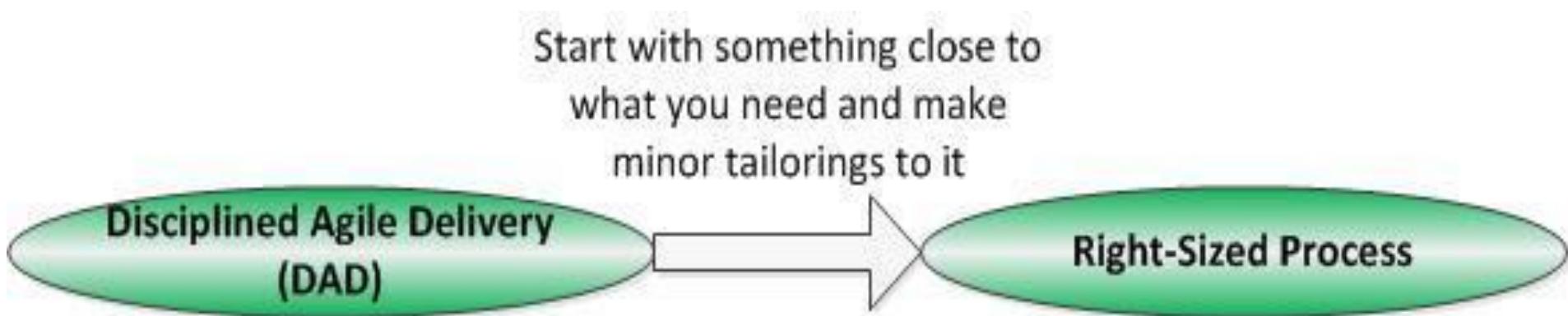
Disciplined Agile Delivery



Copyright 2015 Disciplined Agile Consortium.



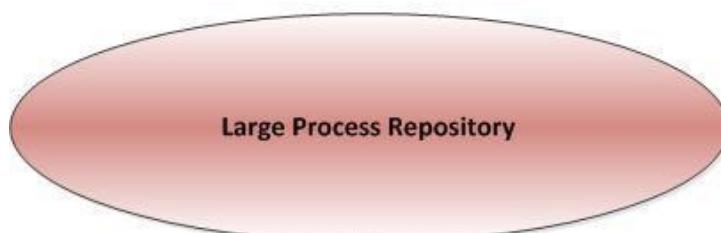
Disciplined Agile Delivery



Copyright 2015 Disciplined Agile Consortium



Process Framework Scope



Starting Point

Rational Unified Process (RUP)

Typical Result

- Teams don't follow the process, and often fake the artifacts to make it appear so
- Management believes there is a well defined process in place that teams are following

Cause

- There is far too much material, and teams often don't know where to start

Tailor down the process to meet your team's actual needs

Right-Sized Process

Disciplined Agile Delivery (DAD)

Add techniques and strategies to meet your team's actual needs

Small Method

Scrum

- "RUPifall" strategy where the RUP phases are tailored to look like waterfall phases
- Your team starts, and ends, with a strategy that is much heavier than it needs to be
- Expensive and time-consuming effort to tailor the process

- The material all sounds like good ideas
- Your team often doesn't have the agile process expertise required to tailor the framework effectively

- Your team starts with a strategy that is much closer to what they need
- Can be overwhelming for people who are looking for a "silver bullet"

- DAD starts with an end-to-end delivery lifecycle that shows how everything fits together
- The goal-driven strategy provides straightforward tailoring guidance

- "Water-Scrum-Fall" strategy where Scrum Construction phase is sandwiched between heavy project initiation and release processes
- Expensive and time consuming way to learn agile

- Your team often doesn't have the agile process expertise to choose (or even know about) the approaches that are best suited for them

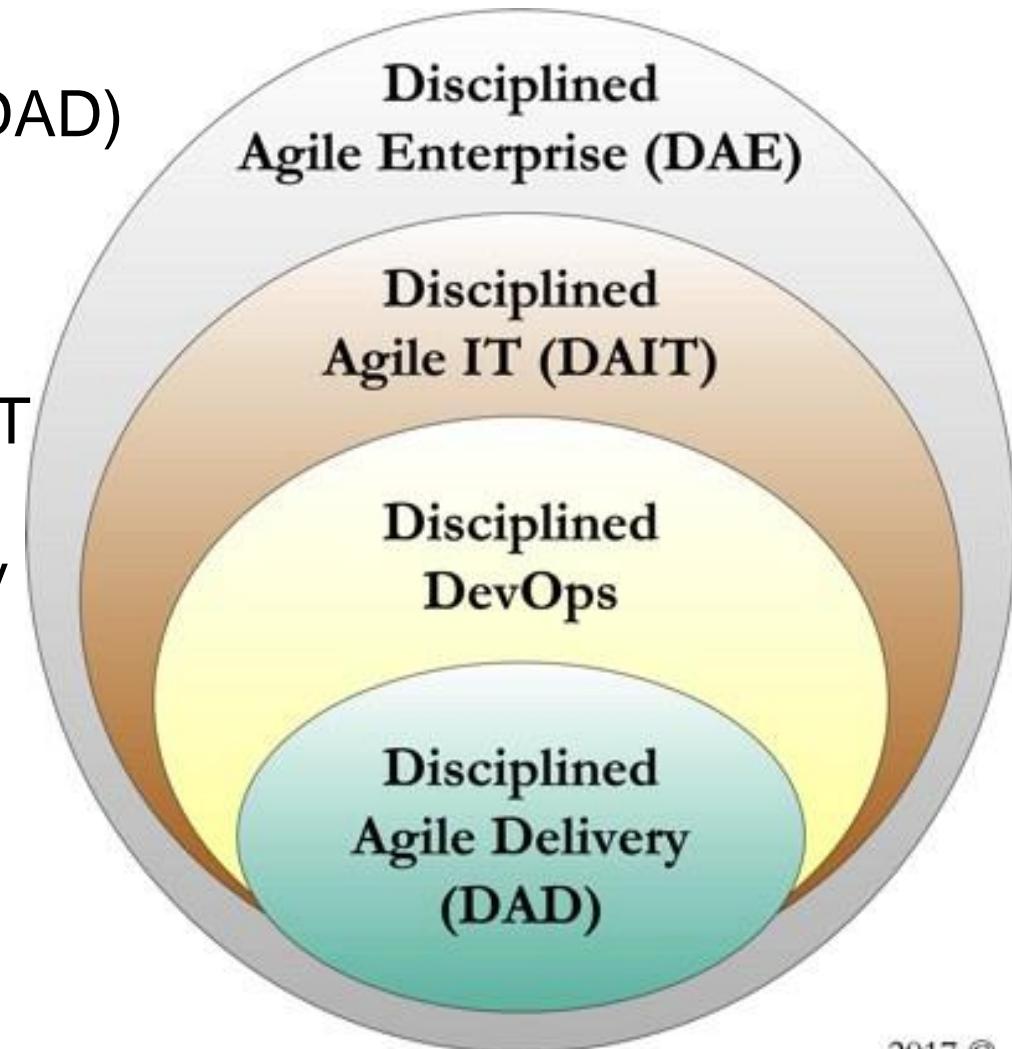
- Cargo-cult agile where teams adopt management practices by not the technical practices

- Overly-simplified, prescriptive strategies

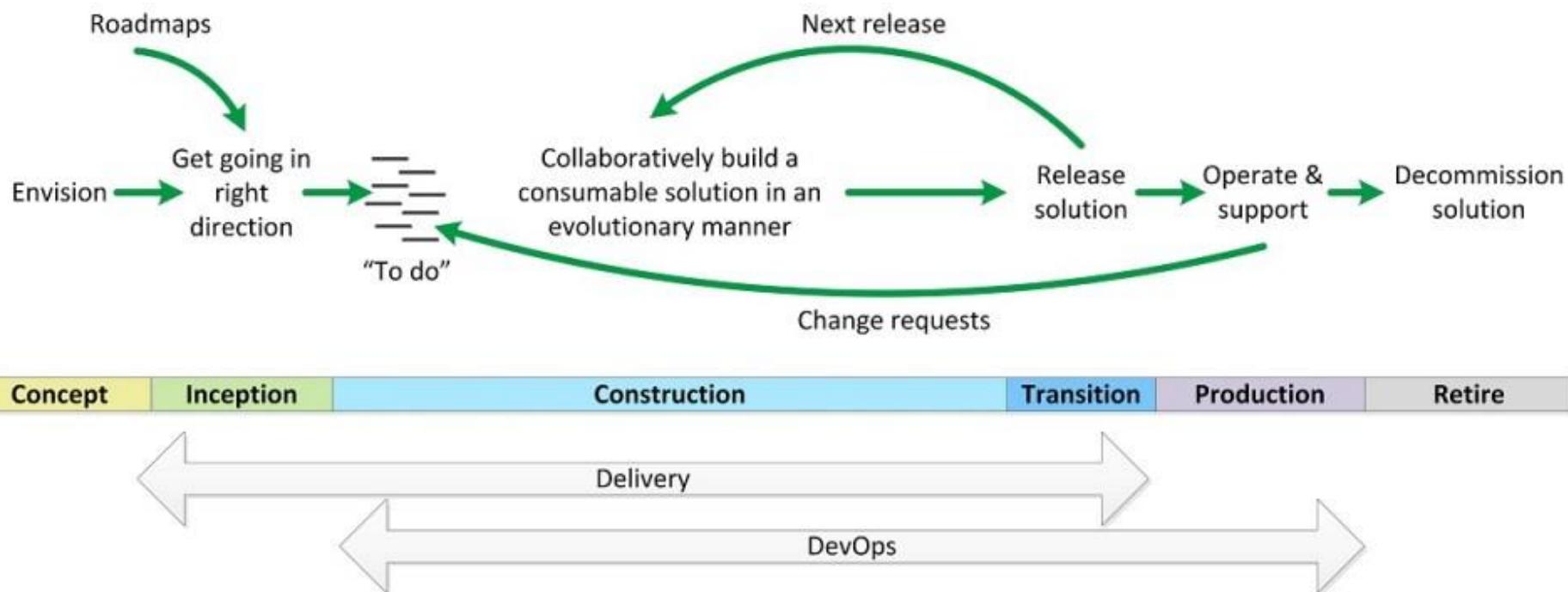
DA framework

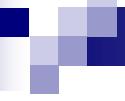
Disciplined Agile Delivery (DAD) is:

- a people-first, learning-oriented
- hybrid agile approach to IT solution delivery
- it has a risk-value delivery lifecycle,
- is goal-driven,
- is enterprise aware,
- is scalable.



DA Lifecycle Overview

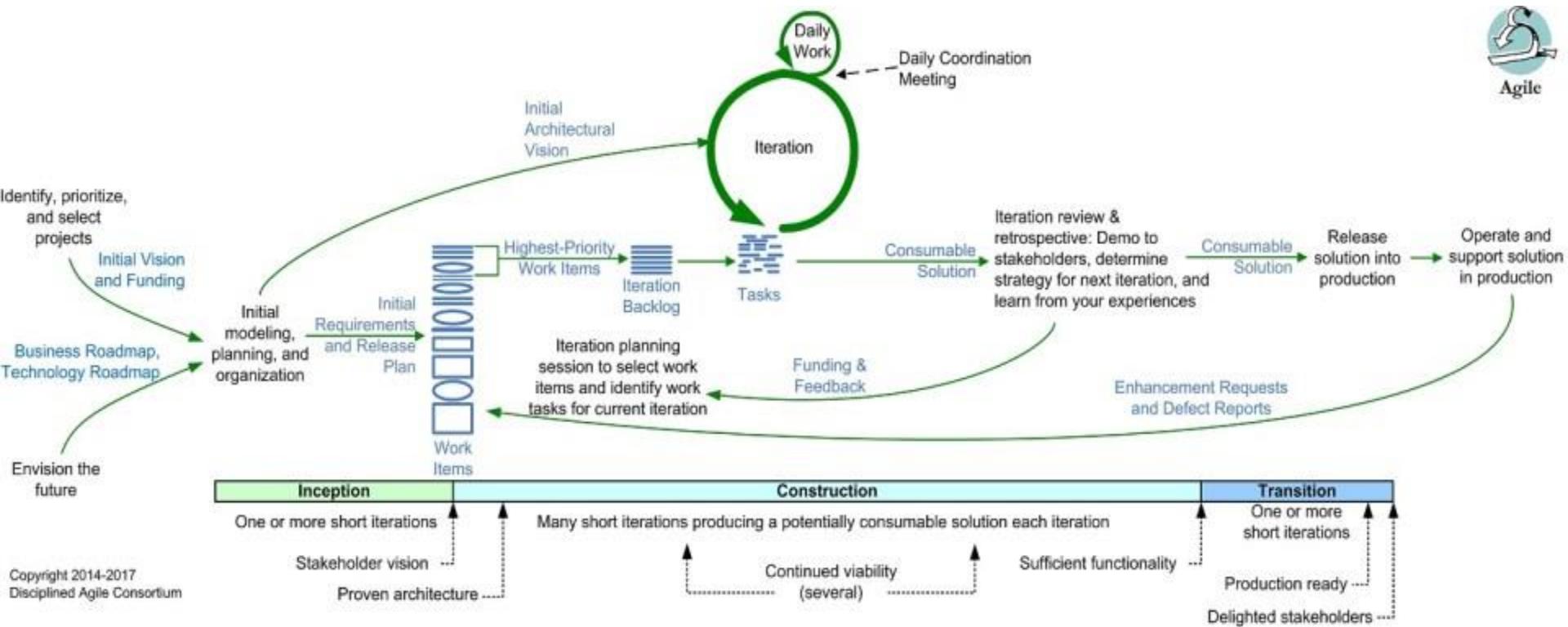




Several lifecycle options

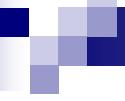
- The Agile/Basic Lifecycle: Extending Scrum
- The Lean/Advanced Lifecycle
- The Continuous Delivery: Agile Lifecycle
- The Continuous Delivery: Lean Lifecycle
- The Exploratory (Lean Startup) Lifecycle

The Agile/Basic Lifecycle: Extending Scrum



Extending Scrum

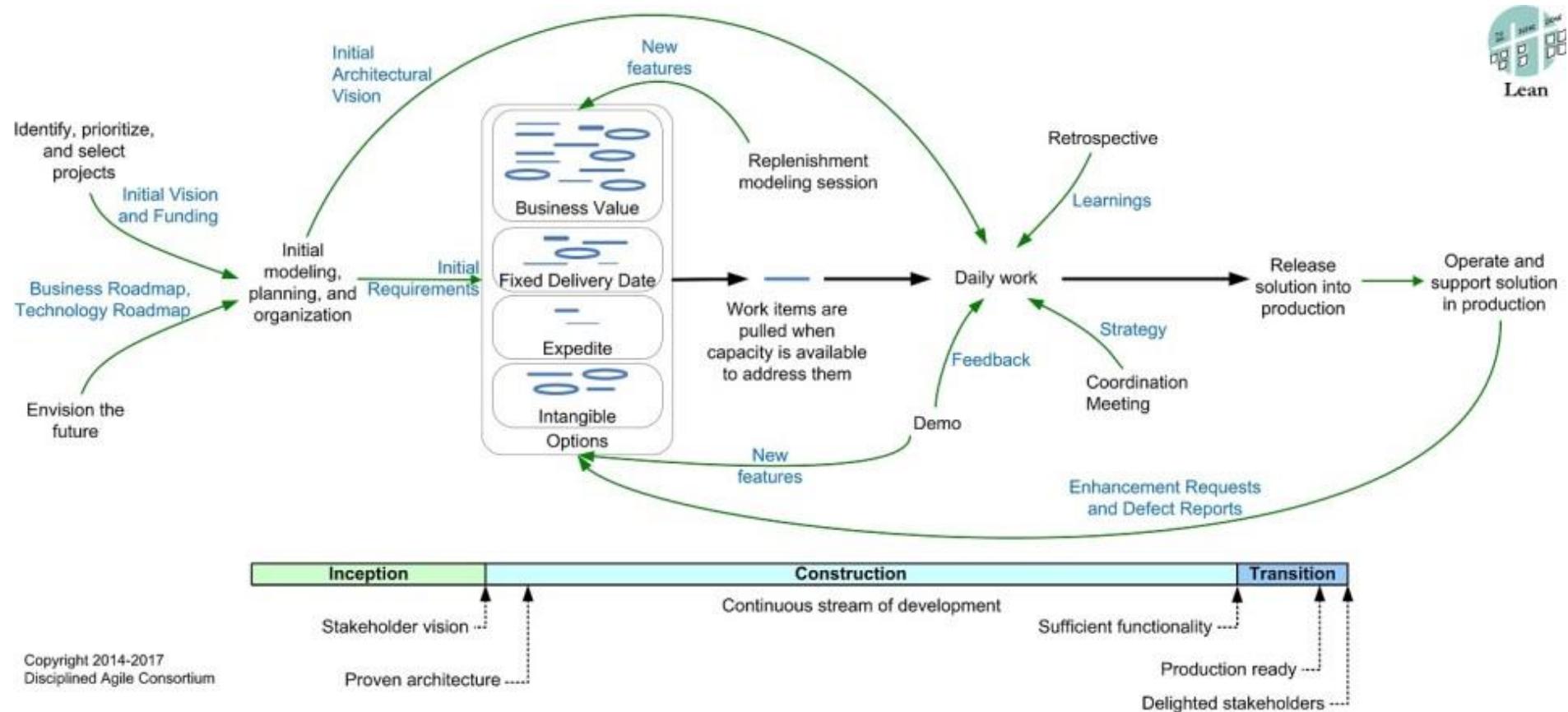
- It's iteration based
- It uses non-Scrum terminology
- It shows inputs from outside the delivery lifecycle
- There is a work item list, not a product backlog
- It includes explicit milestones



When to apply

- The work is primarily enhancements or new features
- The work can be identified, prioritized, and estimated early in the project
- A good choice for new agile teams
- The team is familiar with Scrum and XP
- The team is typically working on a project

The Lean/Advanced Lifecycle



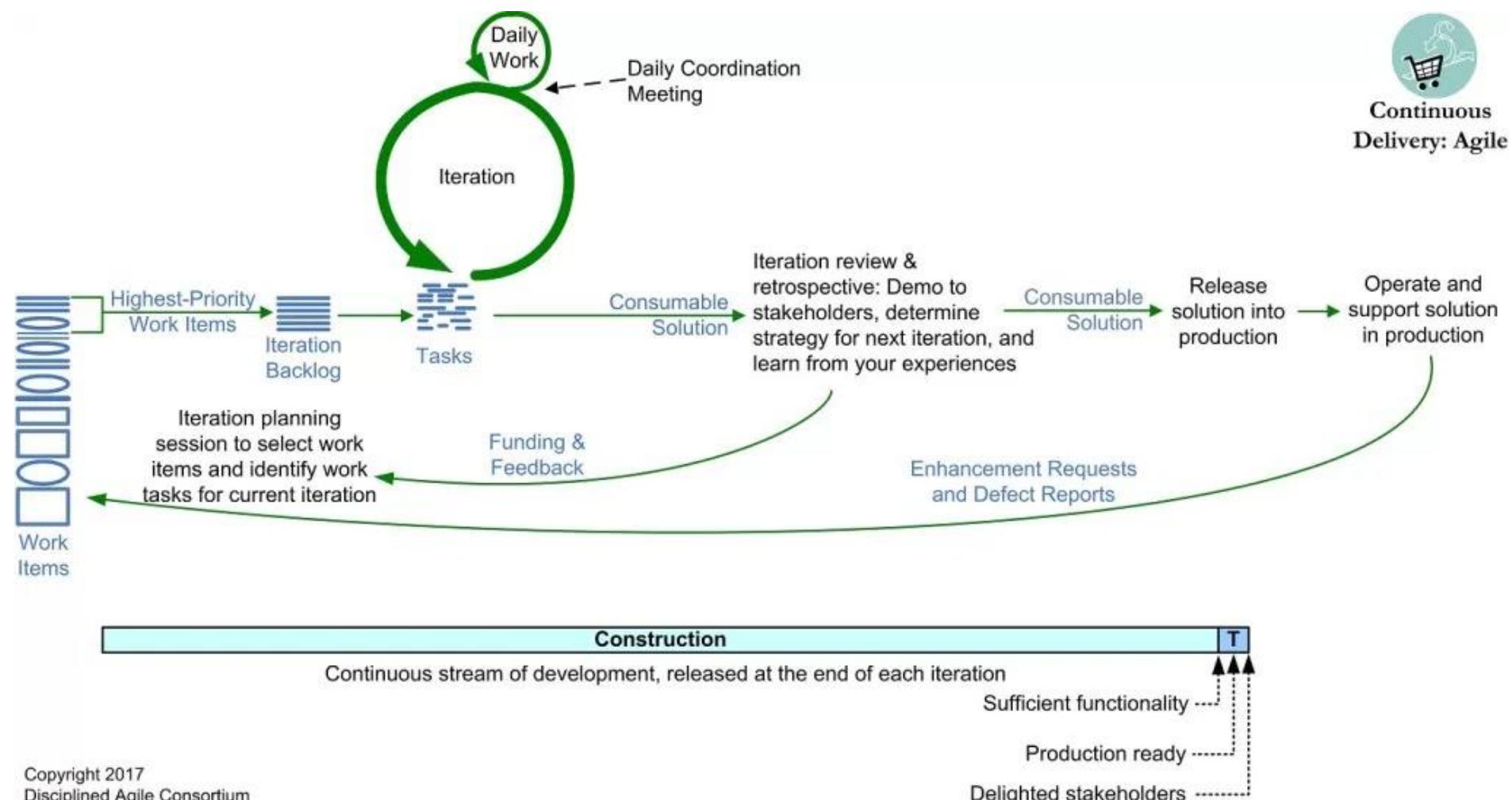
Lean Lifecycle

- It supports a continuous flow of development
- Practices are on their own cadences
- It has a work item pool

When to apply

- Work can be broken down into very small work items of roughly the same size
- Work is difficult to predict in advance. For example, teams that are focused on fixing defects or handling support issues are good candidates for this lifecycle
- The team favors the lean approach of minimizing batch size (which helps to reduce work in progress) and any planning in advance of doing the work
- The team is typically working on a project

The Continuous Delivery: Agile DAD Lifecycle



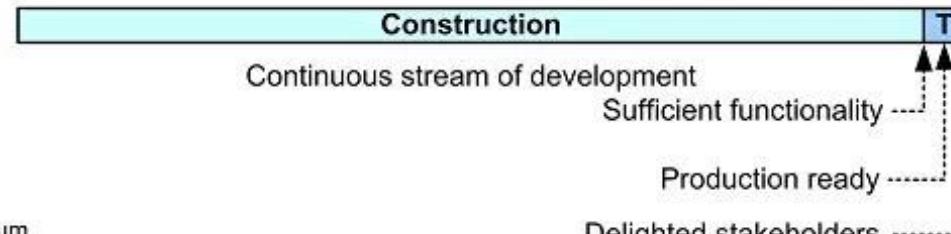
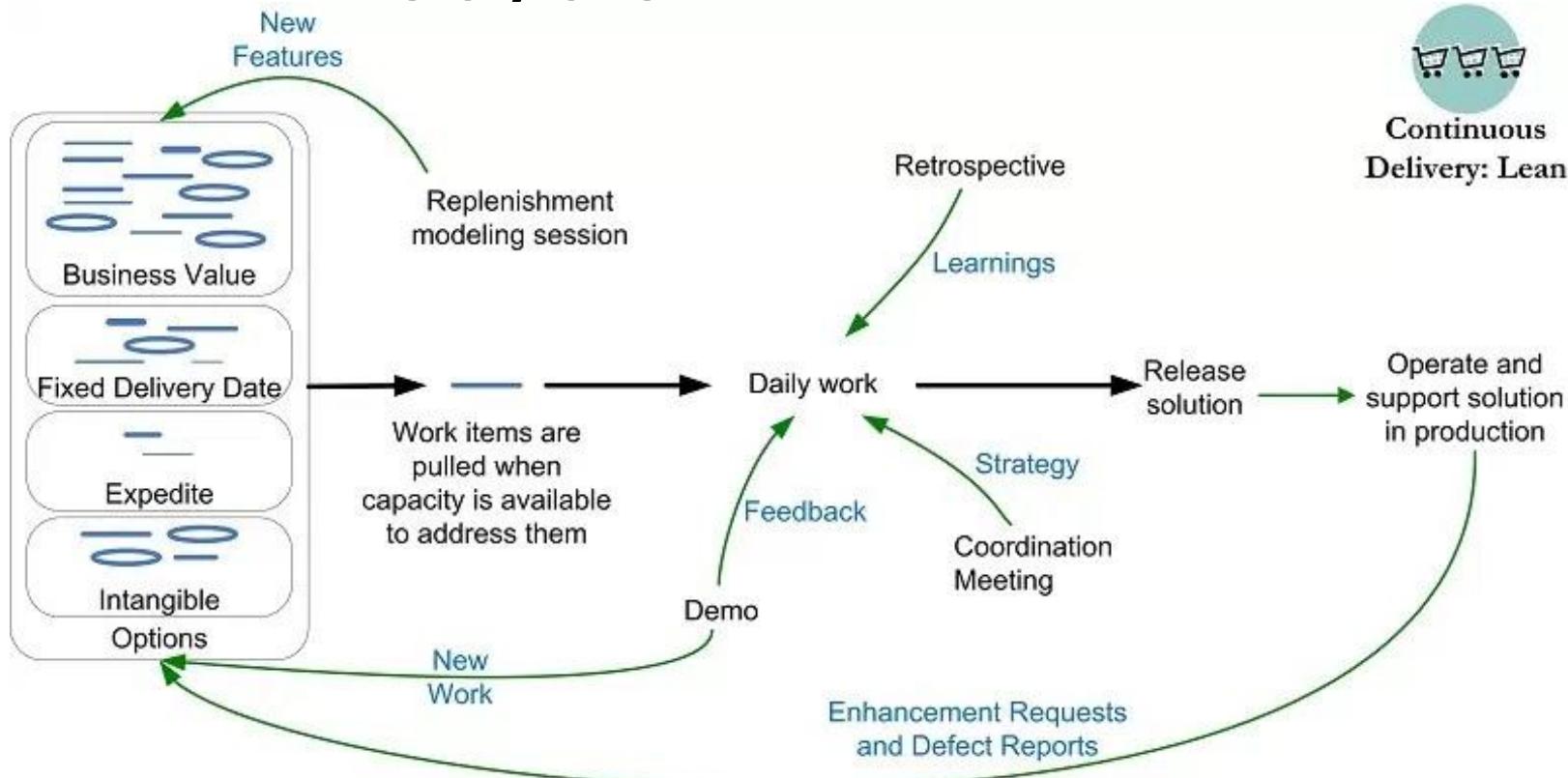
The Continuous Delivery: Agile

- A natural progression from the Agile/Basic lifecycle
- Key difference between this and the Agile lifecycle is in a release of new functionality at the end of each iteration rather than after a set of iterations.
- Teams require a mature set of practices around continuous integration and continuous deployment and other Disciplined DevOps strategies

When to apply

- Solutions that can be delivered to stakeholders in a frequent and incremental basis
- Work remains relatively stable within an iteration
- Organizations with streamlined deployment practices and procedures
- Projects where getting value into the hands of stakeholders rapidly, before the entire solution is complete, is critical
- Teams have mature DevOps practices in place including continuous integration, continuous deployment, and automated regression testing
- The team is long-lived (stable), working on a series of releases over time

The Continuous Delivery: Lean DAD Lifecycle



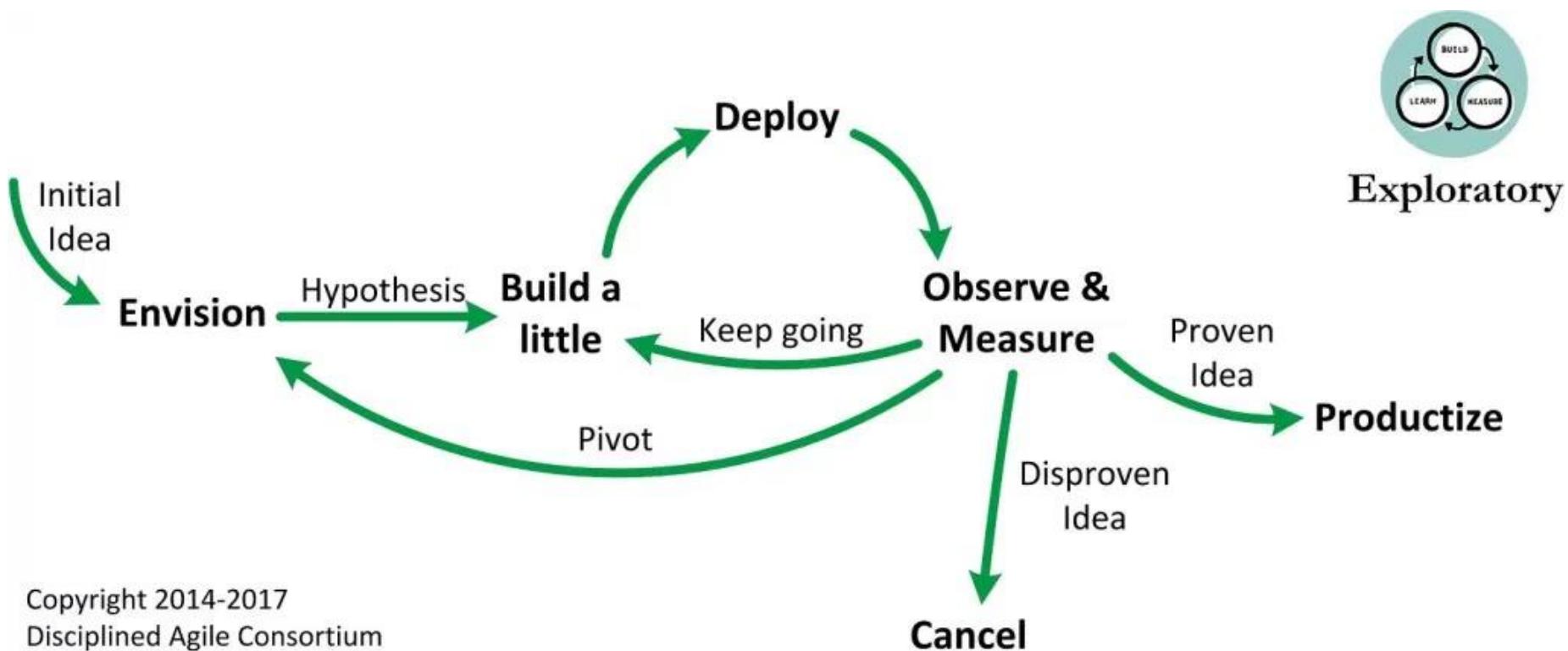
Continuous Delivery: Lean

- A natural progression from the Advanced/Lean lifecycle
- Delivers increments of the solution in a more frequent manner than the other lifecycles
- Requires a mature set of practices around continuous integration and deployment, suitable technical infrastructure

When to apply

- Solutions that can be delivered to stakeholders in a frequent and incremental basis
- New work, including both new requirements and defect reports, arrives often
- Organizations with streamlined deployment practices and procedures
- Projects where getting value into the hands of stakeholders rapidly, before the entire solution is complete, is critical
- Teams with mature DevOps practices in place including; continuous integration, continuous deployment, and automated regression testing
- The team is long-lived (stable), working on a series of releases over time

The Exploratory (Lean Startup) Lifecycle

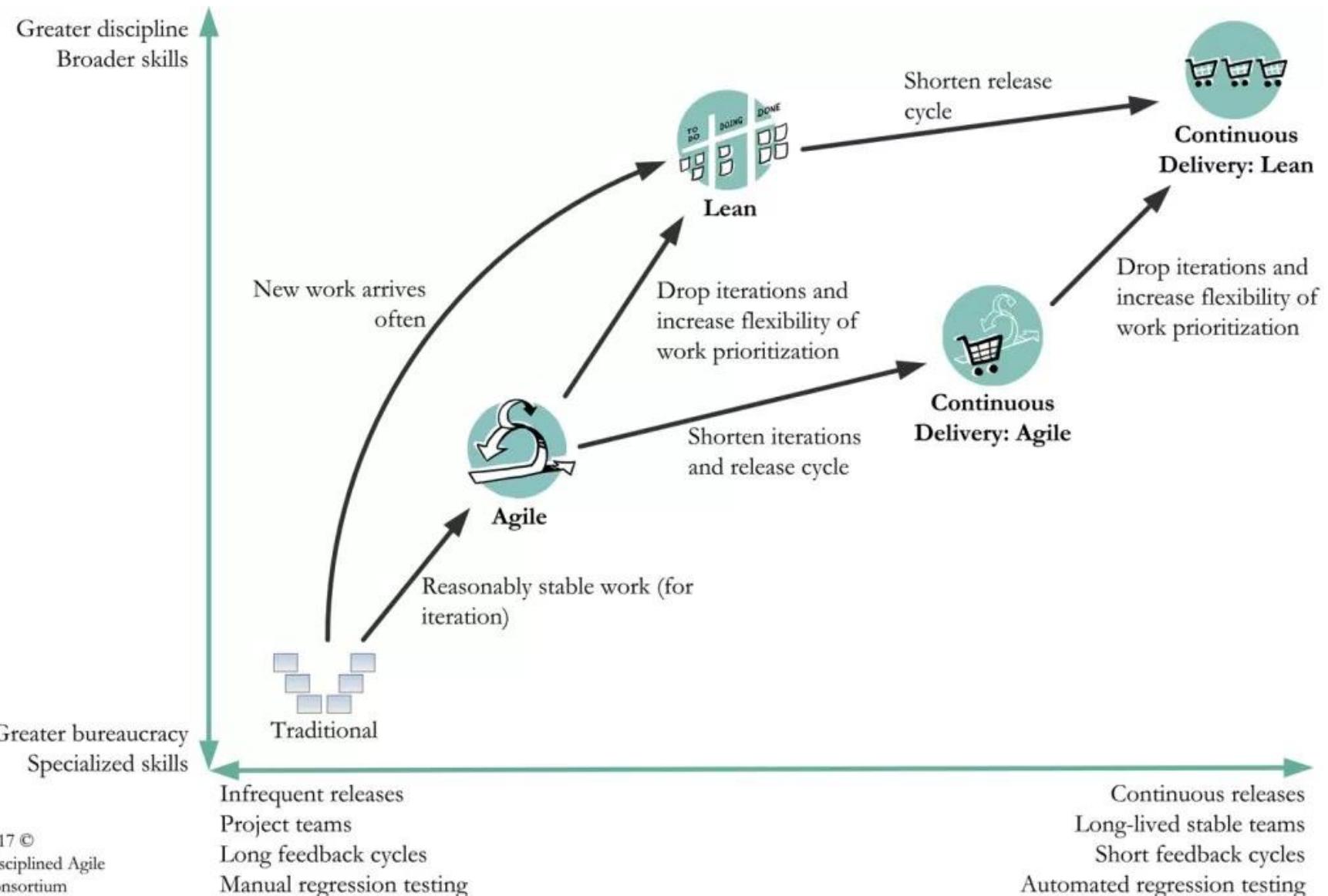


Copyright 2014-2017
Disciplined Agile Consortium

When to apply

- The solution addresses high incertitude cases such as a new unexplored market or a new product
- The stakeholders and delivery team are very flexible in adapting the solution as it is being developed
- You have a valid hypothesis/strategy to test with clear go/no-go criteria for when the test is over
- You are willing to experiment and evolve your idea based on your learnings

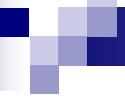
Summary of DAD



Lifecycle	Team Type	Time to Market	Advantages		When to Use
			Disadvantages		
Agile/Basic	Project	Medium	<ul style="list-style-type: none">Straightforward lifecycle based on Scrum that is easy to learn due to its prescription	<ul style="list-style-type: none">Iterations (sprints) motivate teams to build functionality in multi-week batchesReleases into production typically months apartTends to fall apart when requirements change often	Teams new to agile
Lean/Advanced	Project	Fast	<ul style="list-style-type: none">Functionality is released into production when it's ready to goWork can be prioritized via a variety of criteriaSmall batches of work lead to quick flow	<ul style="list-style-type: none">Requires greater skill and discipline compared to the Agile lifecycle	Disciplined teams with quickly evolving requirements
Continuous Delivery: Agile	Product (long lived)	Fast	<ul style="list-style-type: none">Functionality is released into production regularly at a steady flow (typically weekly)	<ul style="list-style-type: none">Requires significant skill and disciplineRequires automated testing, integration, and deployment	Long-running teams
Continuous Delivery: Lean	Product (long lived)	Very Fast	<ul style="list-style-type: none">Functionality is released into production continuously, typically one or more times a day	<ul style="list-style-type: none">Requires significant skill and disciplineRequires automated testing, integration, and deployment	Long-running, disciplined teams

Some Case Studies (optional)

- Extending the Rational Unified Process with a User Experience Discipline: a Case Study
- Change Management Process on Database Level within RUP Framework,
- Key Lessons from Tailoring Agile Methods for Large-Scale Software Development



Quiz time

- Let's switch to Moodle

Next time Discussion topic

- Give an example of a subprocess/discipline/artifact that is instantiated based on specific attribute(s) of a project.
- You may use the above mentioned case studies as examples or your experience.
- “Ex. for my diploma project I will not use a User experience Guideline because it is a research project and the UI is not relevant.”



The project structure (Scope Management)

Lecture 5

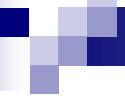
Topic – Scope Management and Work planning

■ Planning the Development

- WBS
- Backlogs

■ References

- Software Project Management: A Unified Framework*, Walker Royce, Addison Wesley
- Fundamentals of Project Management*, James P. Lewis, AMACOM Books
- Software Project Management*, Bob Huges, Mike Cotterrel
- PMBOK
- The Agile Coach <https://www.atlassian.com/agile>



Recap

- Software Development Planning

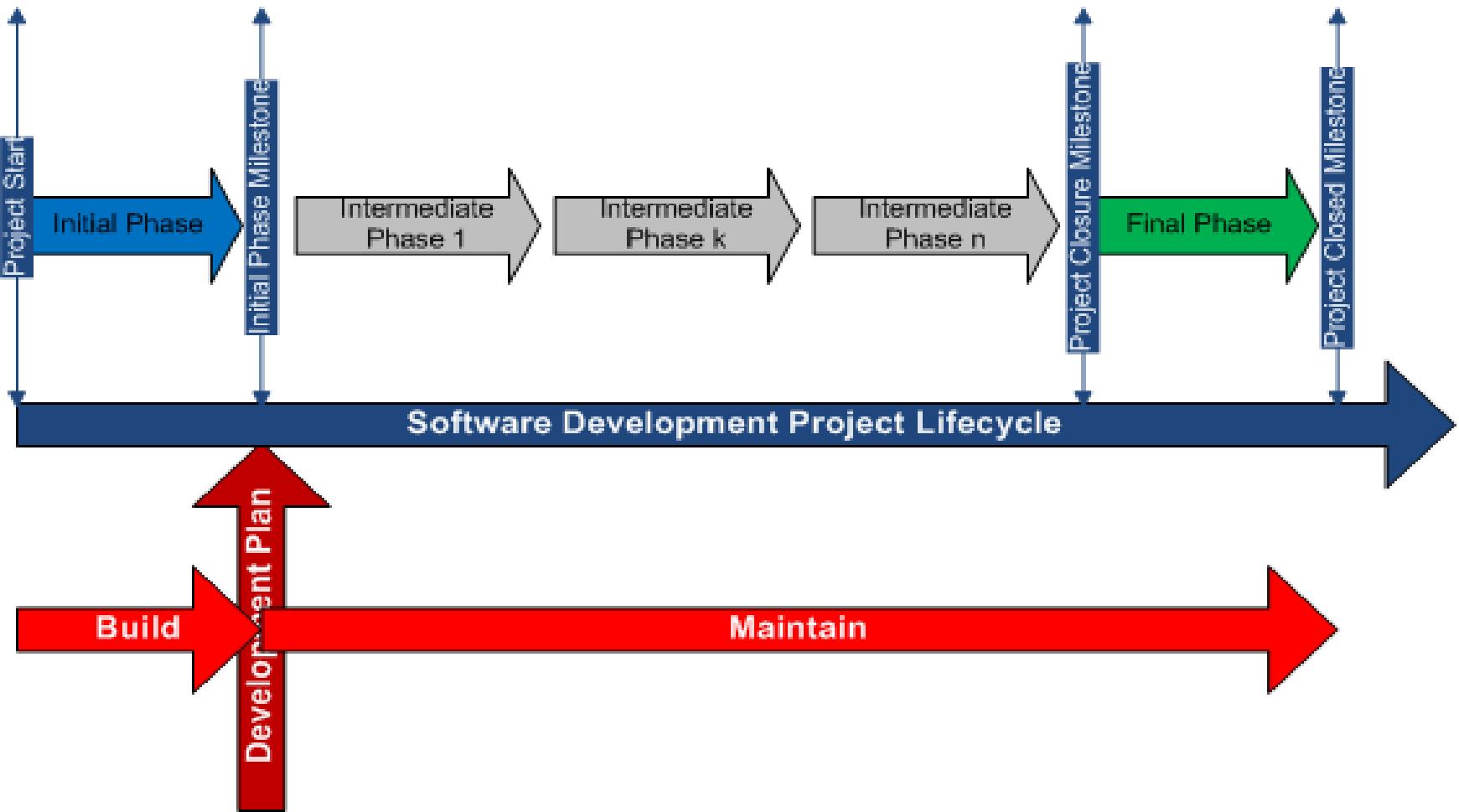
Recap – What IS or ISN'T a Project Plan

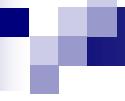
- A project plan IS a collection of different views of the project
- A project plan IS a model of the project's future
- A project plan IS a living artifact
- A project plan is NOT a GANTT chart
- A project plan is NOT a schedule

Recap - PMBOK Definition

- The process necessary for defining, preparing, integrating and coordinating all subsidiary plans into a project management plan.
- The project management plan becomes the primary source of information for how the project will be planned, executed, monitored and controlled, and closed.

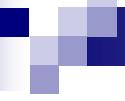
Recap - Planning timeline





Project Planning Process Areas

- Developing the Project Plan
- Interacting with Stakeholders
- Getting Commitment to Plan
- Maintaining the Plan



Specific Goals

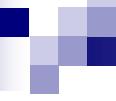
- Establish Estimates
- Develop a Project Plan
- Obtain Commitment to the Plan

Project Planning Roles and Responsibilities

	Responsible	Accountable	Consulted	Informed
Project Management	X	X		
Project Stakeholders			X	X
Project Team	X		X	
Project Review Authority				X

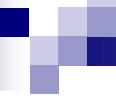
Going further into Planning

- Planning is about answering some questions like:
 - “What must be done?”
 - “When will it be done?”
 - “Who will build it?”
 - “How much will it cost?”



Planning the “WHAT”

- Planning the “what” is crucial
- One frequent reason projects fail is that a significant part of the work is forgotten!



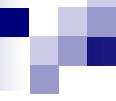
Project Scope Management

- Scope Planning
- Scope Definition
- Create WBS
- Scope Verification
- Scope Control



Scope Planning

- Defining and managing the scope is key to project success
- Detailed project scope statement process
- Process for creating the WBS from the detailed statement
- Formal verification specification
- Change control process



Scope Definition

- Detailed project scope statement
- Change Requests
- Project Scope Management Plan update

Scope Definition [1]

- Detailed project scope statement
 - Project Objectives (Specific, Measurable, Attainable, Relevant, Time-limited)
 - Requirements
 - Boundaries
 - Deliverables
 - Acceptance Criteria
 - Constraints
 - Assumptions

Scope Definition [2]

- Detailed project scope statement
 - Initial Organization
 - Initial Risk
 - Milestones
 - Fund limitations
 - Cost estimate
 - Configuration Management Requirements
 - Approval Requirements

Example

Create WBS

- Project Scope Statement Updates
- Work Breakdown Structure
- WBS Dictionary
- Scope Baseline
- Project Scope Management Plan Updates
- Change Requests

Scope Verification

- Assures that
 - the detailed scope definition,
 - WBS and
 - WBS dictionaryare formally reviewed and approved by the project stakeholders

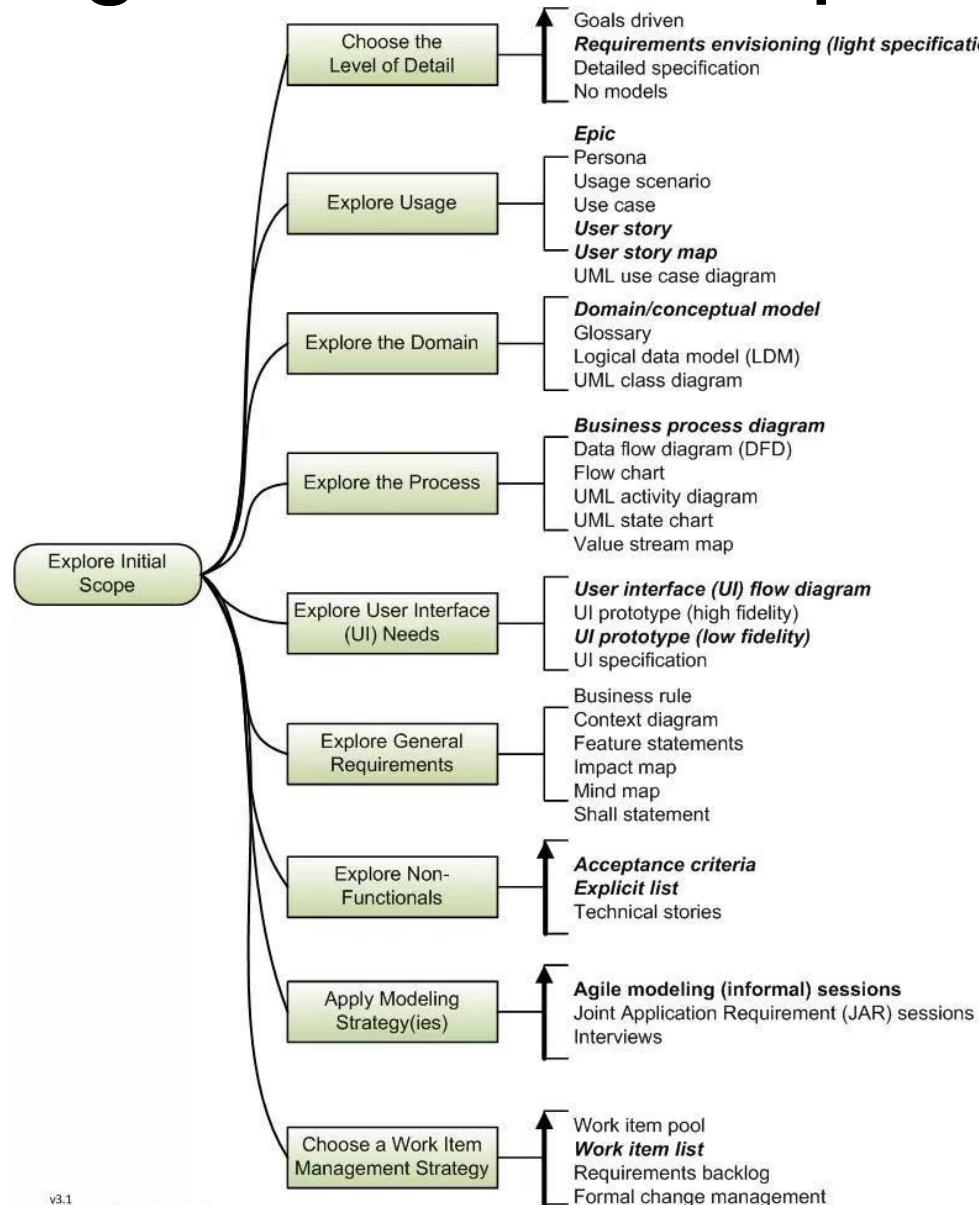
Scope Control

- Assures that all Requested Changes and Corrective Actions are processed by the Integrated Change Control Management process
- => avoid Scope Creep

Scope Creep

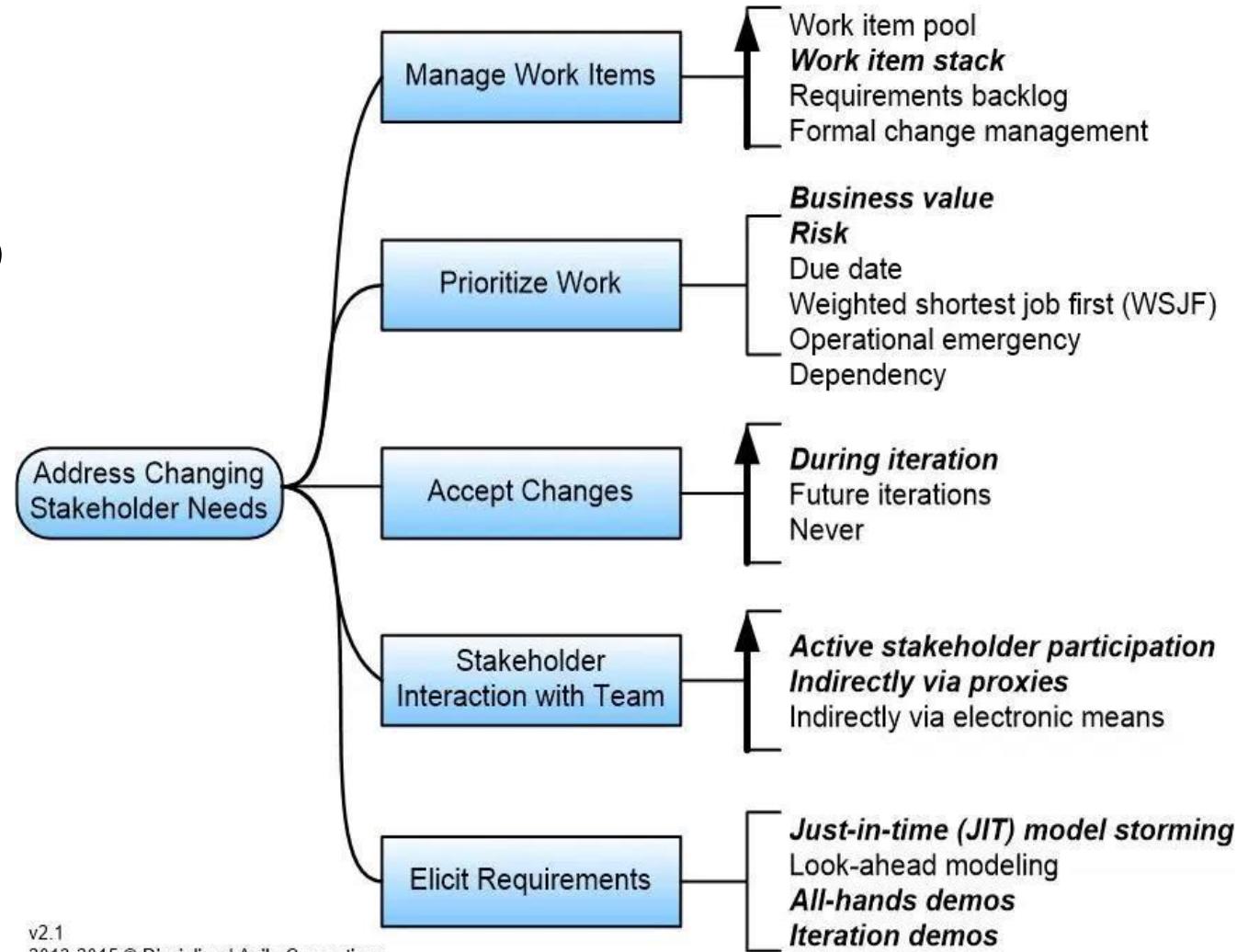
- “adding features and functionality (project scope) without addressing the effects on time, costs, and resources, or without customer approval” [PMBOK]

Exploring the Initial Scope with DAD



Address changing needs

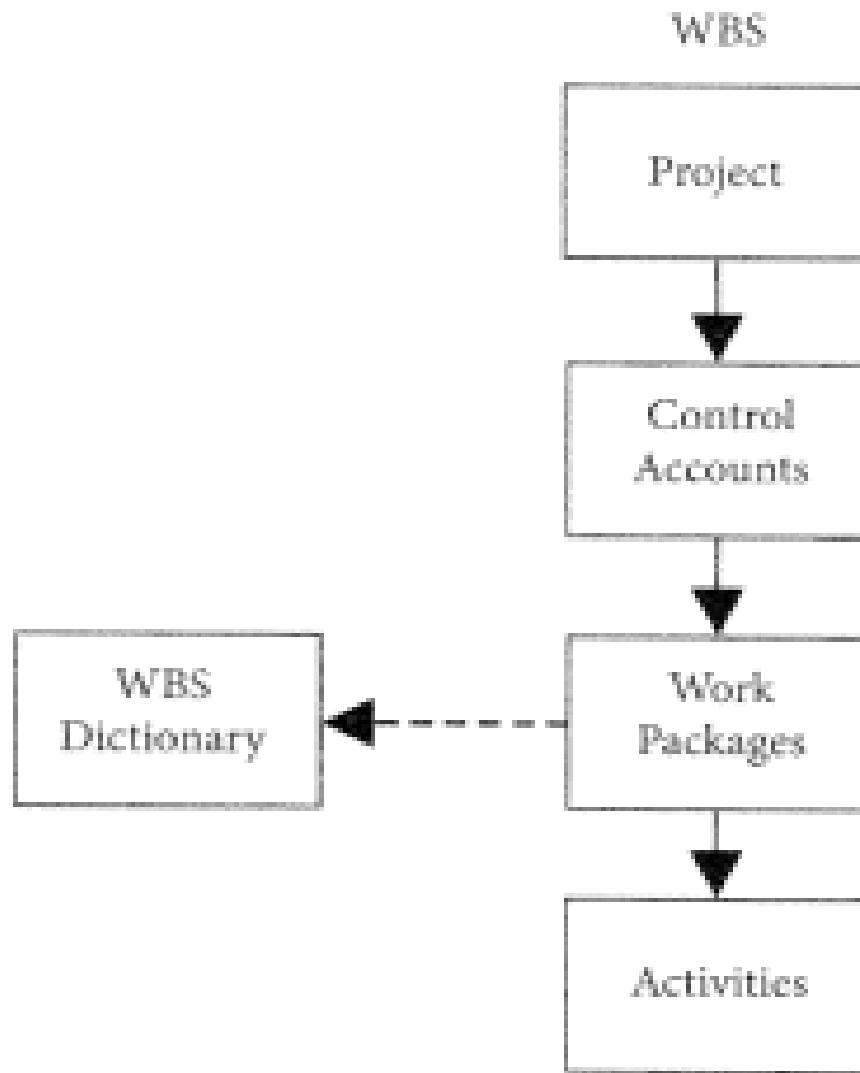
■ Mitigate Scope Creep



What is the WBS?

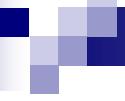
- WBS is a hierarchical decomposition of work
- Can be action-driven (verbs) or deliverable-driven (nouns)
- Organizes and defines the total scope of the project
- Each level of the WBS represents an increased level of detail

WBS Introduction



WBS Dictionary example

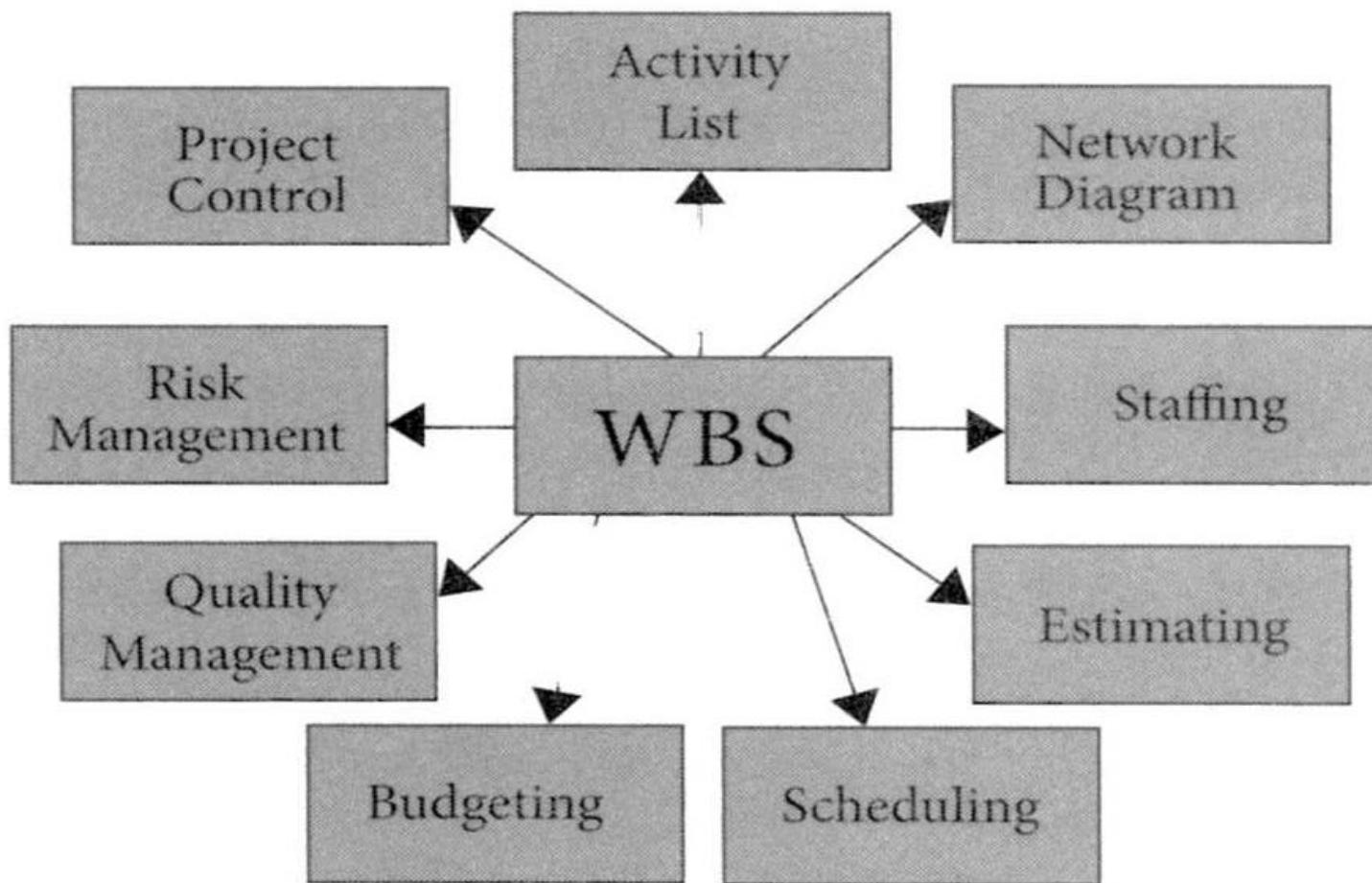
WBS #:	1.1.1	Task:	Create Plan
Est. Level of Effort:	40 hrs	Owner:	Project Manager
Resources Needed:	Subject Matter Experts	Work Products:	MS Project Plan
Description of Task:	Development of a detailed project plan that lists all key resources, tasks, milestones, dependencies, and durations.		
Input:	<ul style="list-style-type: none">Approved Project CharterSMEs		
Dependencies:	<ul style="list-style-type: none">Approval of Budget		
Risk:	<ul style="list-style-type: none">Changes to IT Apps plans and deliverablesIT Apps implementation releases, which conflict with implementation		
WBS #:	1.1.2	Work Item:	Make Budget
Est. Level of Effort:	16 hrs	Owner:	Project Manager
Resources Needed:	CFO, CIO, Executive Sponsor	Work Products:	ITPR
Description of Task:	Development and documentation of the project budget based on plan and resources.		
Input:	<ul style="list-style-type: none">Approved Project CharterSMEs		
Dependencies:	<ul style="list-style-type: none">Approval of Project Charter		
Risk:	<ul style="list-style-type: none">Changes to IT Apps plans and deliverablesIT Apps implementation releases which conflict with implementation		



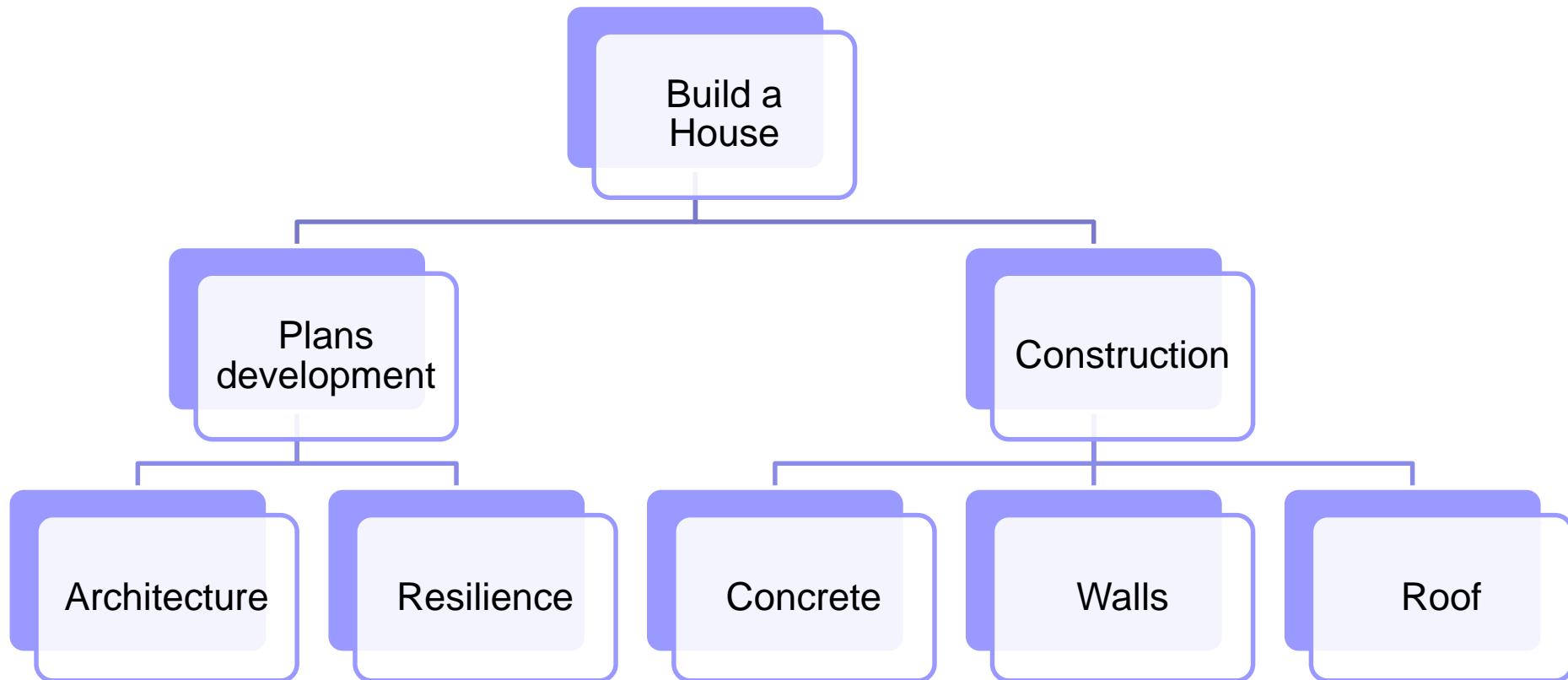
WBS Goals

- Improve estimates
- Keep the team focused
- Assign work to resources
- Keep the project on track

WBS



WBS Example

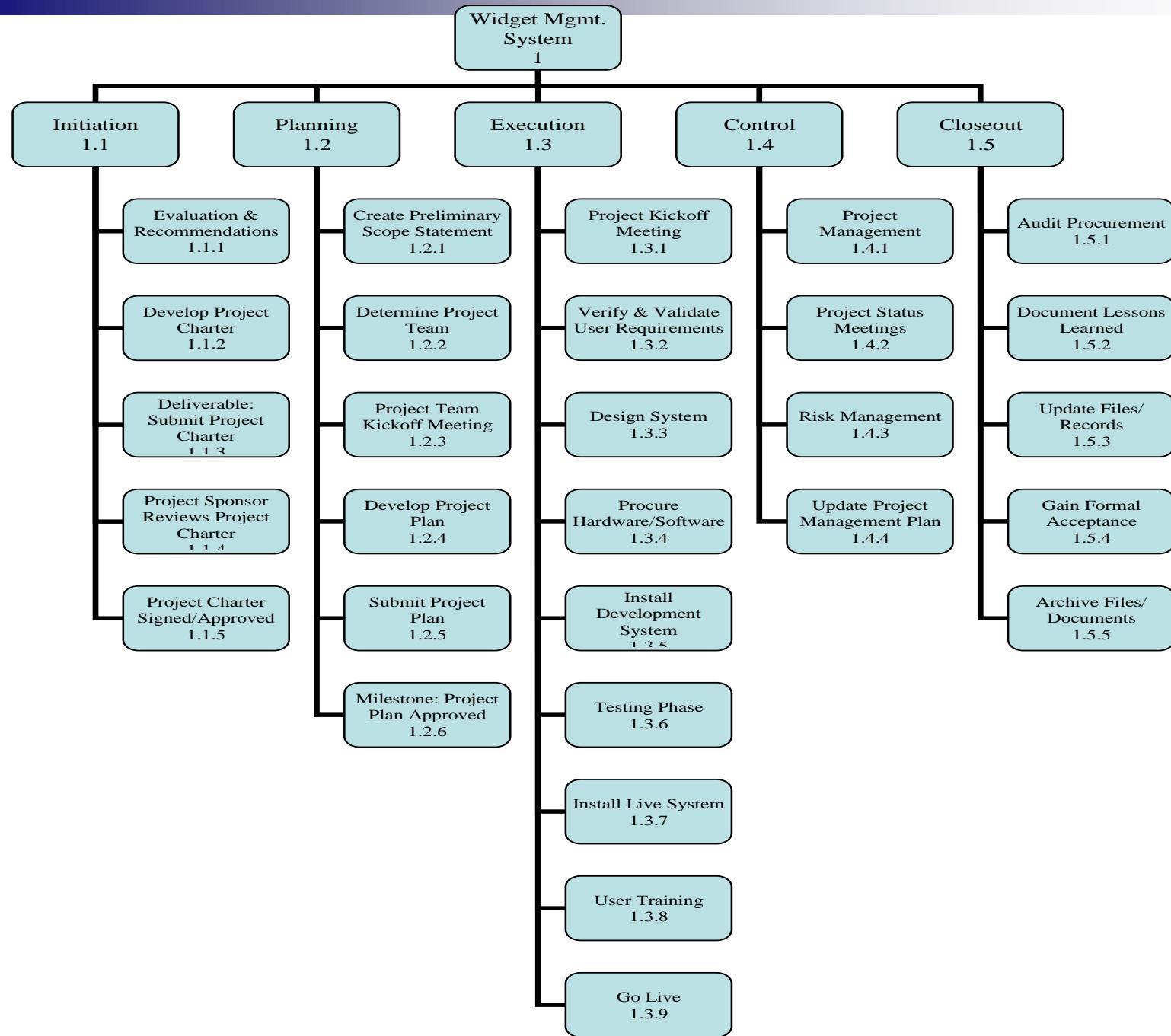


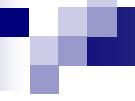
WBS Example

Level 1	Level 2	Level 3
1 Widget Management System	1.1 Initiation	1.1.1 Evaluation & Recommendations 1.1.2 Develop Project Charter 1.1.3 Deliverable: Submit Project Charter 1.1.4 Project Sponsor Reviews Project Charter 1.1.5 Project Charter Signed/Approved
	1.2 Planning	1.2.1 Create Preliminary Scope Statement 1.2.2 Determine Project Team 1.2.3 Project Team Kickoff Meeting 1.2.4 Develop Project Plan 1.2.5 Submit Project Plan 1.2.6 Milestone: Project Plan Approval



Level 1	Level 2	Level 3
1 Widget Management System	1.3 Execution	1.3.1 Project Kickoff Meeting 1.3.2 Verify & Validate User Requirements 1.3.3 Design System 1.3.4 Procure Hardware/Software 1.3.5 Install Development System 1.3.6 Testing Phase 1.3.7 Install Live System 1.3.8 User Training 1.3.9 Go Live
	1.4 Control	1.4.1 Project Management 1.4.2 Project Status Meetings 1.4.3 Risk Management 1.4.4 Update Project Management Plan
	1.5 Closeout	1.5.1 Audit Procurement 1.5.2 Document Lessons Learned 1.5.3 Update Files/Records 1.5.4 Gain Formal Acceptance 1.5.5 Archive Files/Documents



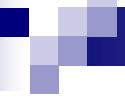


Some WBS Issues

- Prematurely structured around the product design
- Prematurely decomposed, planned and budgeted into either too much detail or too little detail
- Is project specific, and cross project comparison is impossible

Evolutionary WBS

- An evolutionary WBS should organize around the PROCESS framework rather than around the product framework
- Example (what the book says)
 - First-level WBS elements are Disciplines
 - Second-level elements are lifecycle phases
 - Third-level should focus on the activities that produce the artefacts of every phase/iteration

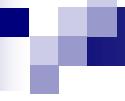


WBS tailoring criteria

- Project scale
- Organizational structure
- Custom development
- Business context
- Precedent experience

WBS RACI

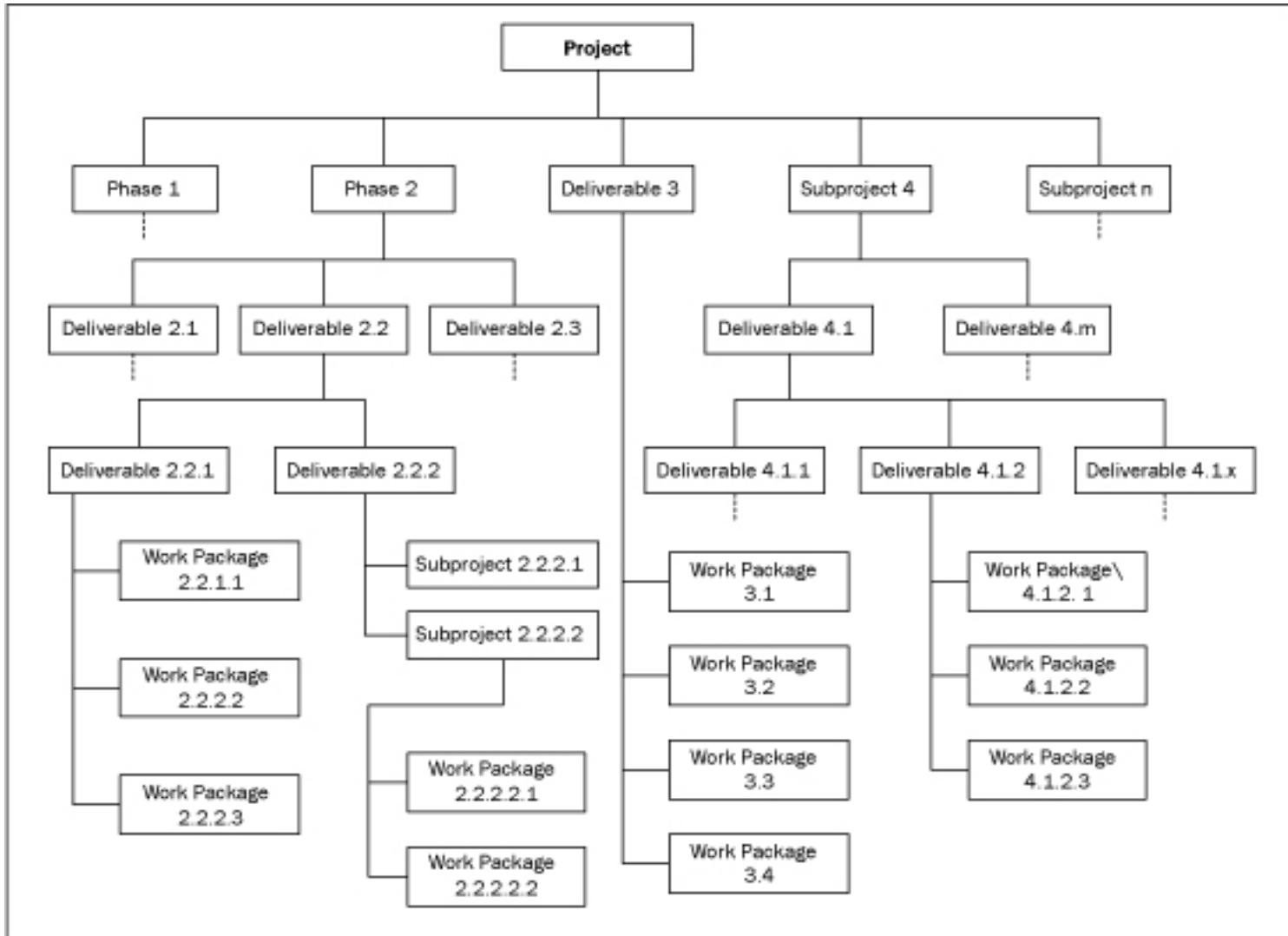
	Responsible	Accountable	Consulted	Informed
Project Manager	X	X		
Project Team	X			
Project Review Authority				X
Project Stakeholders			X	X



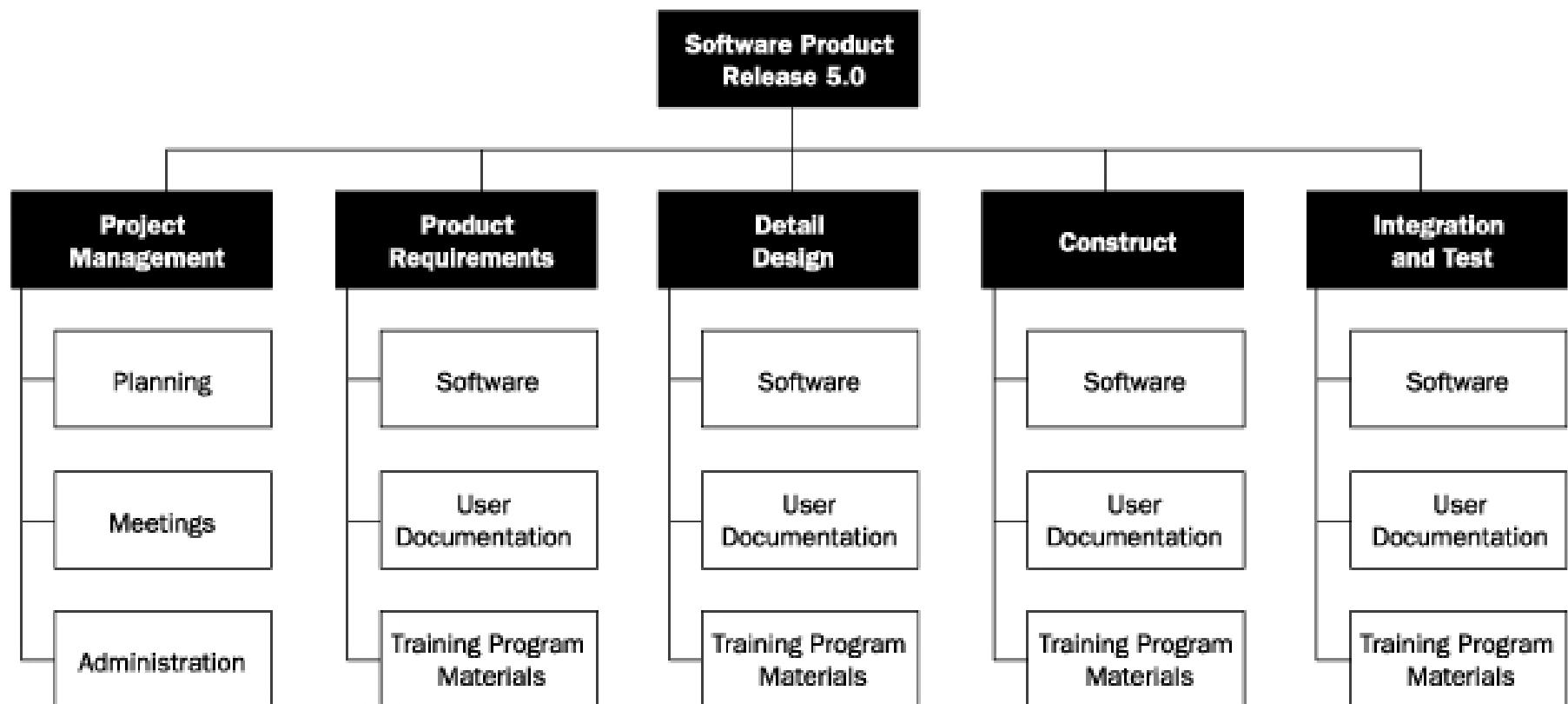
WBS Creation Activities

- Identifying the deliverables and related work
- Structuring and organizing the WBS
 - Top-down approach
 - Chronological approach
 - Bottom-up approach
- Verifying that the degree of decomposition of the work is necessary and sufficient.

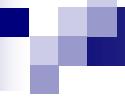
WBS Structure by Phase



WBS By Discipline

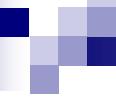


The WBS is illustrative only. It is not intended to represent the full project scope of any specific project, nor to imply that this is the only way to organize a WBS on this type of project.



Project Plan Architecture

- WBS is the project's plan “architecture”
 - It must encapsulate change
 - It must evolve with the appropriate level of detail through the project lifecycle
 - It must cover ALL project tasks, and NO MORE

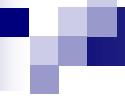


WBS development guidelines

- What is the best level of detail?
- When to stop?
- Whom should be involved?
 - See the RACI table

Level of detail

- Projects can under-plan as they can over-plan
 - Balance is crucial in finding the right level of detail
 - Must be detailed enough to buy in stakeholders and still remain manageable
 - How many levels?
 - Normally 3-7 levels

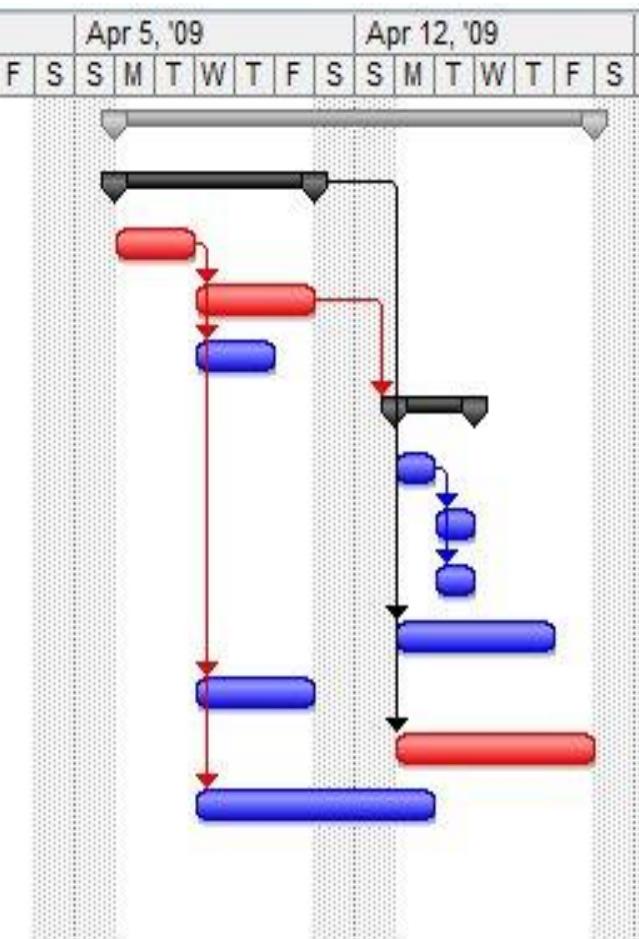


Is it a work package?

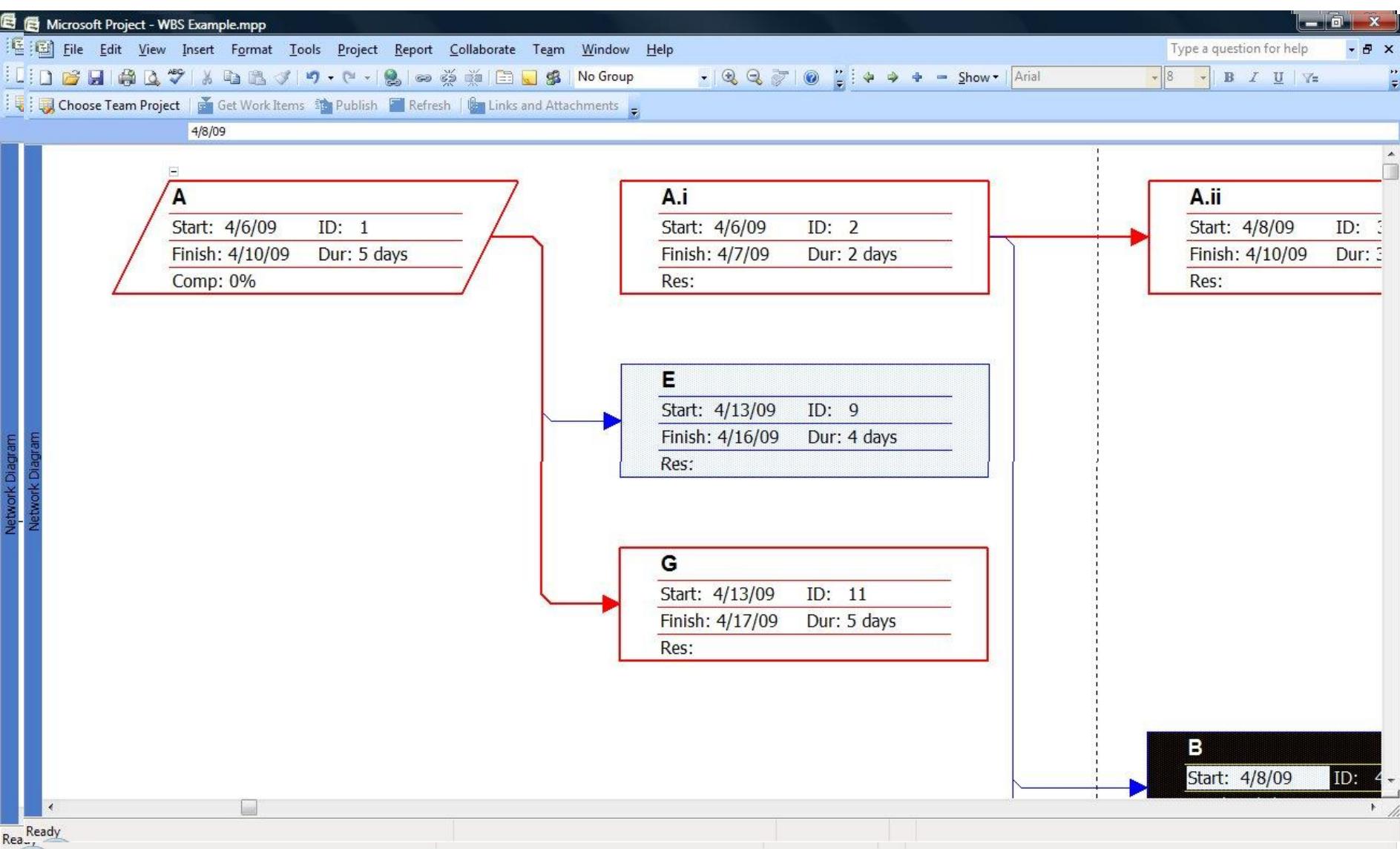
- If you can estimate the work
- If it takes between 8 and 80 hours
- If you can complete it without interruption
- If you can outsource it

GANTT (MS Project)

WBS	Task Name	Duration	Start Slack	Start	Finish	Prede
0	- WBS Example	10 days	0 days	Mon 4/6/09	Fri 4/17/09	
1	- A	5 days	0 days	Mon 4/6/09	Fri 4/10/09	
1.1	A.i	2 days	0 days	Mon 4/6/09	Tue 4/7/09	
1.2	A.ii	3 days	0 days	Wed 4/8/09	Fri 4/10/09	2
2	B	2 days	6 days	Wed 4/8/09	Thu 4/9/09	2
3	- C	2 days	3 days	Mon 4/13/09	Tue 4/14/09	3
3.1	C.i	1 day	3 days	Mon 4/13/09	Mon 4/13/09	
3.2	C.ii	1 day	3 days	Tue 4/14/09	Tue 4/14/09	6
3.3	C.iii	1 day	3 days	Tue 4/14/09	Tue 4/14/09	6
4	E	4 days	1 day	Mon 4/13/09	Thu 4/16/09	1
5	F	3 days	5 days	Wed 4/8/09	Fri 4/10/09	2
6	G	5 days	0 days	Mon 4/13/09	Fri 4/17/09	1
7	H	4 days	4 days	Wed 4/8/09	Mon 4/13/09	2

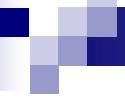


Network Diagrams



Generic WBS

- Project WBS Template
- Software Development.mpp
- MSF Application Development.mpp

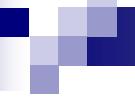


Project organization

- Closely linked to WBS
- Two organizations
 - Matrix
 - Project
- Plan for evolution and accommodate change

Project organization key points

- Organizational structures **form the architecture of the teams**
- Organizations involved in software line of business **need to support a common process**
- Project organizations need to **allocate artefacts and responsibilities clearly and balanced across project team(s)**
- The project organization **must evolve with the WBS and the project lifecycle**



■ What about defining work in Agile?

Roadmaps

- A plan of action for how a product or solution will evolve over time
- What are the relative priorities of each initiative?
- When do we intend to work on each initiative?
 - Are there particular dates the team needs to hit?
 - What dependencies does the program have—either internal, or on other teams

Roadmaps

- Which teams are working on each initiative?
 - Do the current teams have availability in their schedules and enough capacity?
 - Can we keep the current agile teams stable?
 - If not...How will teams be re-organized?

Roadmap

Next-gen project Software project

Roadmap Backlog Board Reports

Give feedback Learn more

Roadmap

Status Today Months

Epic	OCT	NOV
Build Mobile App	2018/10/15 – 2019/01/14	
Add booking functionality		
Improve UMUX Lite Score for checko...		
Redundancy plan for outage		
Improve NPS for mobile app users by ...		
Reduce load time to 2.2 seconds		
Custom Filters in app		
Refresh homepage footer		
Create Facebook & Twitter connector		
Standardize fonts in app		
Build checkout experience		
Standardize fonts in app		
+		

ATMT-14 Build Mobile App

Status In Progress

Maggie wants to use Jira on mobile so she can keep up with her team on the go

Child issues

ATMT-48 Add welcome screen for m... TO DO

Linked issues

relates to

ATMT-26 Broken status ind... IN REVIEW

Assignee

Shuri Jones

Labels

None

Add a comment...

ATMT-14 Build Mobile App

Status In Progress

Maggie wants to use Jira on mobile so she can keep up with her team on the go

Child issues

ATMT-48 Add welcome screen for m... TO DO

Linked issues

relates to

ATMT-26 Broken status ind... IN REVIEW

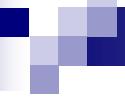
Assignee

Shuri Jones

Labels

None

Add a comment...



Roadmap Anti-patterns

- Future planning is completely ignored
- The "rest of the business" is kept in the dark as to what the team is up to.
- The roadmap is continuously updated (or never updated).
- Detailed requirements are weighing down the roadmap.

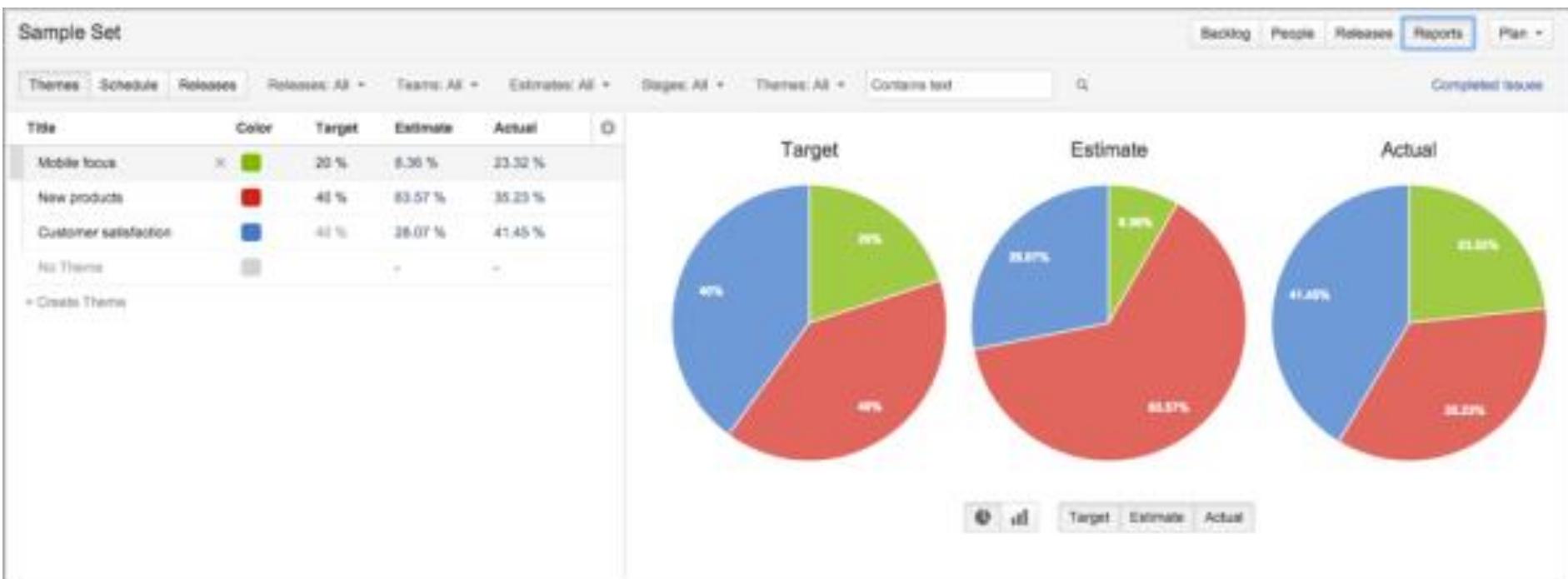
Work decomposition



- **Themes** are large focus areas that span the organization, not time-oriented, focused on relative resource allocation.
- **Initiatives** are collections of epics that drive toward a common goal.
- **Epics** are large bodies of work that can be broken down into a number of smaller tasks (called stories).
- **(User) Stories**, are short requirements or requests written from the perspective of an end user.

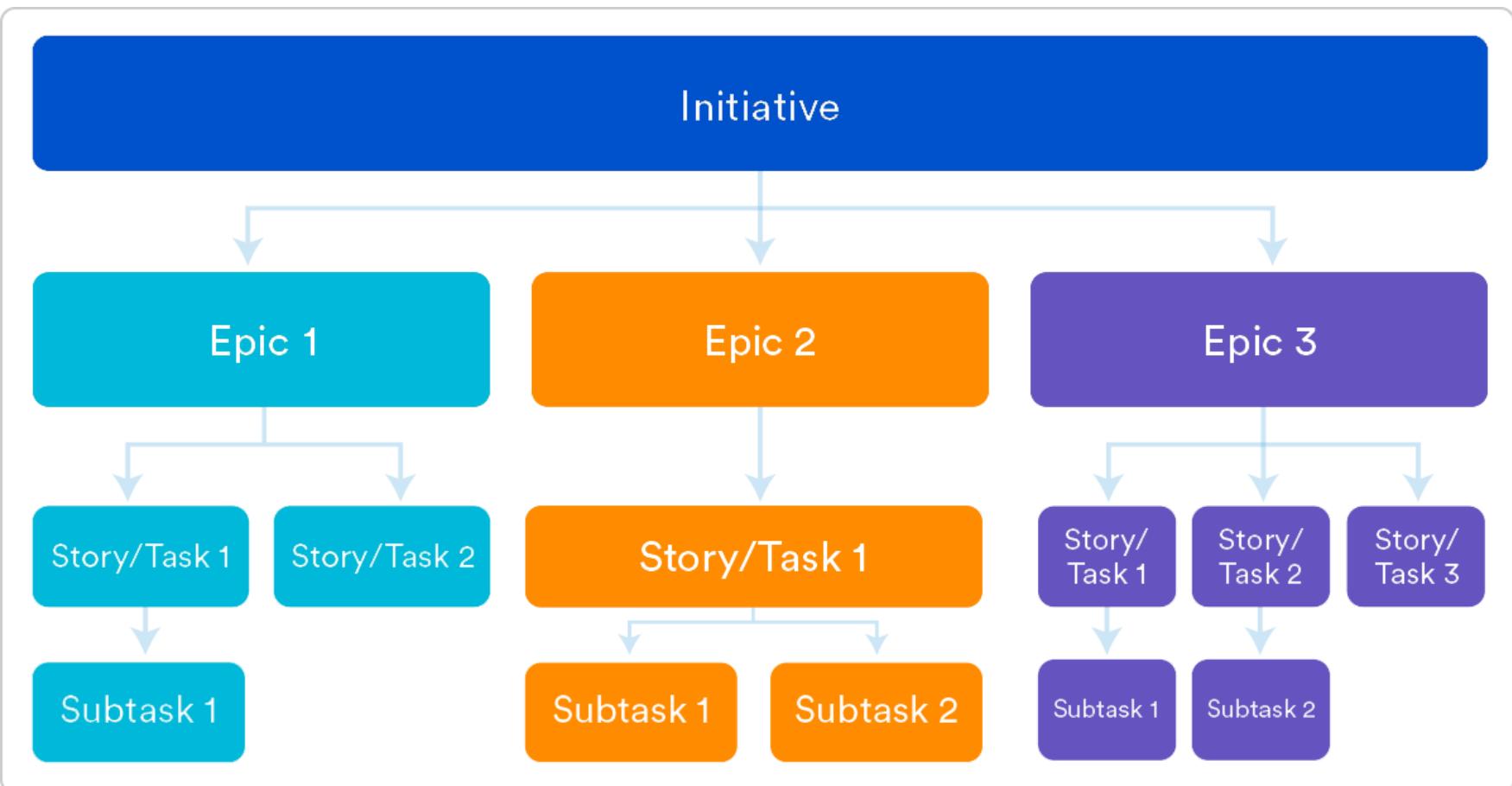
Themes In Jira

- Target – resource allocation
- Estimate - Sums up all backlog items labeled by theme
- Actual - JIRA application issues work logs



Initiatives

- Are collections of epics



Epics and user stories

- Tactical level
- Epics are Multi-Sprint
- User-Stories are Single-Sprint
- Describe work items in a non-technical language and from a user's perspective.
- As a {type of user}, I want {goal} so that I {receive benefit}.

Theme
“Build the promotion platform”

Epic

“As a Marketer, I want to create a promotion so it can be used in a marketing campaign”

Epic

“As a marketer, I want to create a segment so it can be used to target specific audiences”

User Story

As a marketer, I want to enter a discount amount so it can be used when creating a discount promotion

User Story

As a marketer, I want to establish a start and end date for my promotion so I can define validity

User Story

As a marketer, I want to view segment size relative to all consumers, so I can reach a critical mass

User Story

As a marketer, I want to give my segment a name so I can easily access it within the platform

Epic details in Jira

The screenshot shows the Jira interface for managing an epic titled "Frontend redesign". On the left, a sidebar lists the components of the epic: "Database migration", "Frontend redesign", "Stakeholder signoff" (status: DONE), "FY19 planning", "Tech debt", and "Papercuts". A large Gantt chart spans from March to May, showing the duration of each component. The "Frontend redesign" component is the longest, starting in March and ending in May. The "FY19 planning" component starts in April and ends in May. The "Tech debt" component starts in April and ends in May. The "Papercuts" component starts in May. The "Stakeholder signoff" component is completed in March.

PN-19
Frontend redesign

Status
To Do

Add a description...

Child issues

- PN-3 Redesign the home page TO DO
- PN-9 Investigate feasibility of ... IN REVIEW
- PN-20 Checkout page rede... IN PROGRESS

Assignee
Kevin Bui

Labels
None

Start date
2018/03/06

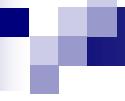
Due date
2018/05/14

Backlog

- Seen by different roles
- As a **team member**, I want to know what user stories to work on next with an understanding of how my work fits into the overall matrix.
- As a **product owner**, I want to see how features are progressing so I can adjust and refine priorities with the team.

Backlog

- As a **project manager**, I want to see how the product develops and how overall progress is going.
- As a **stakeholder**, I want to know when my thing is ready.

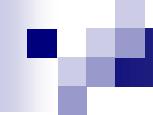


Backlog structures

- **Team Organization:** What are the roles and responsibilities? Who will do what?
- **Process:** What steps add value when we refine an idea toward a user story or requirement?
- **Product Component:** Which part of our product will this change affect? Software, hardware, both?

Backlog structures

- **Planning Horizon:** How far does the backlog stretch in time? The next 4-6 sprints or the whole year?
- **Type of Work:** Do we mix bugs and features? How do we classify different product backlog items?
- **Goals:** What are we trying to achieve? How does each product backlog item contribute to reaching high-level goals?



Team organization

▼ **Area 1** Delegated to: Product Owner

▼ **Team Yellow's Backlog** Delegated to: Team Yellow

User Story 1 

User Story 2 

User Story 3 

► **Team Blue's Backlog** Delegated to: Team Blue

- **Teams can resolve dependencies.** When they can see their backlogs side by side, teams have some visibility they didn't have before
- **User stories can lose the context.** One workaround is to complement this method with heavy use of release tagging for goal setting.
- **Products can take new shapes.** Products tend to take on the characteristics of the organizational structures that creates them.

Process

Funnel

- ▶ **Wish List 1** Delegated to: Stakeholder 1
- ▶ **Wish List 2** Delegated to: Stakeholder 2
- ▶ **Product Backlog** Delegated to: Product Owner
- ▶ **Verification / Release Ready**
- ▶ **Done** Archived

Workflow can be optimized.

Backlog size can be limited. Useful in situations when the product owner is good at saying no, and they want to keep as few items as possible in their backlog.

Stakeholders can participate.

User stories can lose context regarding where they fit into the over-arching picture.

Lean oriented organizations

Components

- Product modularity can be supported.
- Traditional planning methods can excel.
- Overhead can be high. This structure requires a lot of detailed thought and planning up front
- Managing dependencies can be difficult.
- Resembles WBS

▼ Product 1

▼ Component 1

▼ Sub-system 1.1

Change 1

Change 2

► Sub-system 2.1

► Component 2

► Product 2

Planning horizon

- Product Vision
- Product Roadmap
- Release Plan
- Iteration Plan
- Daily Commitment
- **Stakeholders can easily understand it.**
- **Team commitments can be hard to delineate.**

▼ Vision

▼ Roadmap Item 1

▼ Release 1

User Story 1.1 

User Story 1.2 

User Story 1.3 

► Release 2

► Roadmap Item 2

► Roadmap item 3

Type of work

- ▶ **Process Improvements**
- ▶ **Refactoring**
- ▶ **Defects**
- ▶ **New Features**

- **Teams can evenly distribute their attention.** Teams can pick at least one thing from each bucket of different work items. It's also easier to understand work distribution among different types.
- **Classes of Services can be simplified.**
- **Value can become invisible.** Recommendation: make type of work a feature, rather than letting it dominate backlog structure.

Goals

▼ Vision

▼ Goal 1

▼ Feature 1 / Epic 1

User story 1 

User story 2 

► Feature 2 / Epic 2

► Goal 2

- **Teams can stay focused on business value** and improve the right things in the product.
- **Work can be more measurable and accountable.** Quick, measurable feedback loops can be created to understand whether we are working on the right thing or not. Also, we become held accountable for meeting our goals, which we see every day in the backlog.

WBS vs. Backlog

[<http://blog.schauderhaft.de/2016/09/30/workbreakdownstructure-vs-backlog/>]

- “A *hierarchical decomposition of the total scope of work to be carried out by the project team to accomplish the project objectives and create the required deliverables.*”
- “*The Product Backlog comprises an ordered list of requirements that a Scrum Team maintains for a product. It consists of features, bug fixes, non-functional requirements, etc.—whatever must be done to successfully deliver a viable product*”

Common features

- Initially, both are unsorted collections of tasks
- Both live with the project (updated)
- Both get prioritized (on different criteria)
- Both are hierarchically structured
- In the end, both contain every task performed

Differences

- **The WBS doesn't care about value** - A backlog item is done when it can be used as intended.
- **The Backlog is worked on together** - generally, the WBS is worked on only by the Project Manager, or a small team.
- **WBS plans the whole project at the same level of detail** - Backlog elements are planned at different levels of detail (i.e. low-priority items are not planned in detail)

Wrap-up

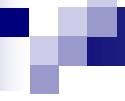
- Use a top-down approach. Begin at the project goal level and break the work successively to lower levels of definition.
- Create it with the input from domain experts and team members.
- Do not include any work which is not part of the project.
- Cover the entire scope of the project at least at the upper level of the WBS

Wrap-up continued

- Each component of the WBS, excepting the highest one, is a part of the parent WBS component.
- The lowest level of the WBS shall have the following properties:
 - It is possible to estimate its need for resources, duration and costs.
 - The start and the end of the execution can be clearly defined.

Wrap-up continued

- It's realization can be outsourced. It has a deliverable or a clear part of it.
- It can be completed within reasonable time limits. There is no universally accepted rule regarding the length of an work package. However, it shall not exceed 1-2 weeks to complete.
- Specific to the software development projects, it can be executed by a single developer.



Quiz Time

- Please switch over to Moodle

Discussion for next time

- Considering your diploma project and the process you defined, construct a 4-level decomposition of work (ex.WBS) that covers the activities you have to do. Try to estimate at least 2 of the leaf-level tasks.

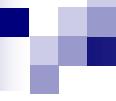


Time Management

Lecture 6

Recap: Going further into Planning

- Planning is about answering some questions like:
 - “What must be done?”
 - “When will it be done?”
 - “Who will build it?”
 - “How much will it cost?”



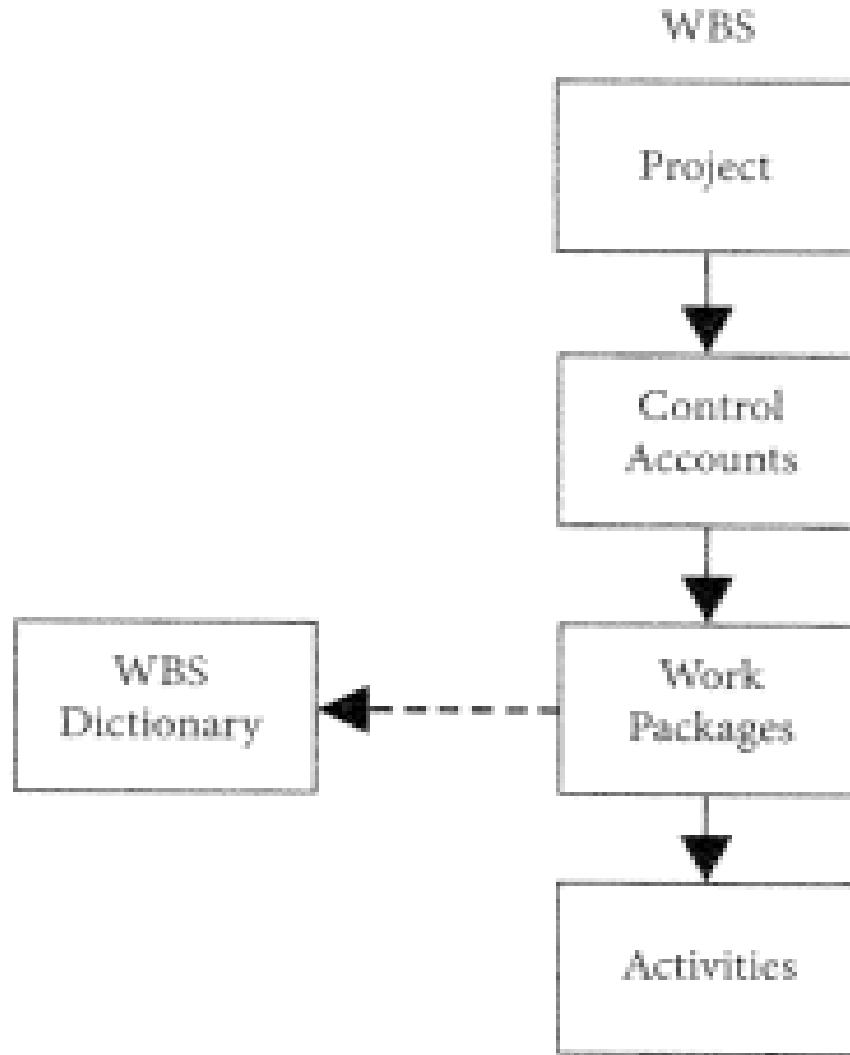
Recap: Planning the “WHAT”

- Planning the “what” is crucial
- One frequent reason projects fail is that a significant part of the work is forgotten!

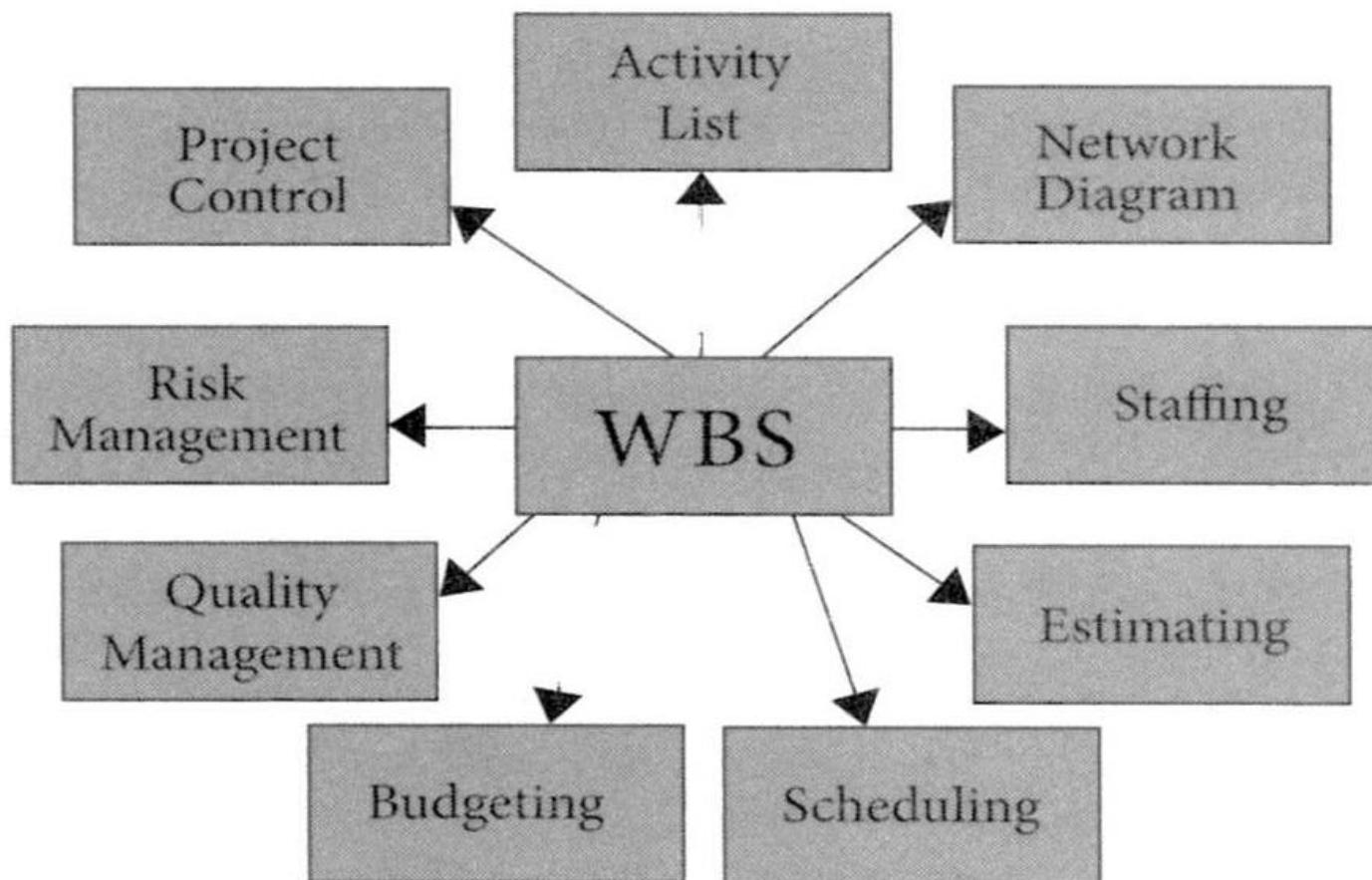
Recap: Project Scope Management

- Scope Planning
- Scope Definition
- Create WBS
- Scope Verification
- Scope Control

Recap: WBS



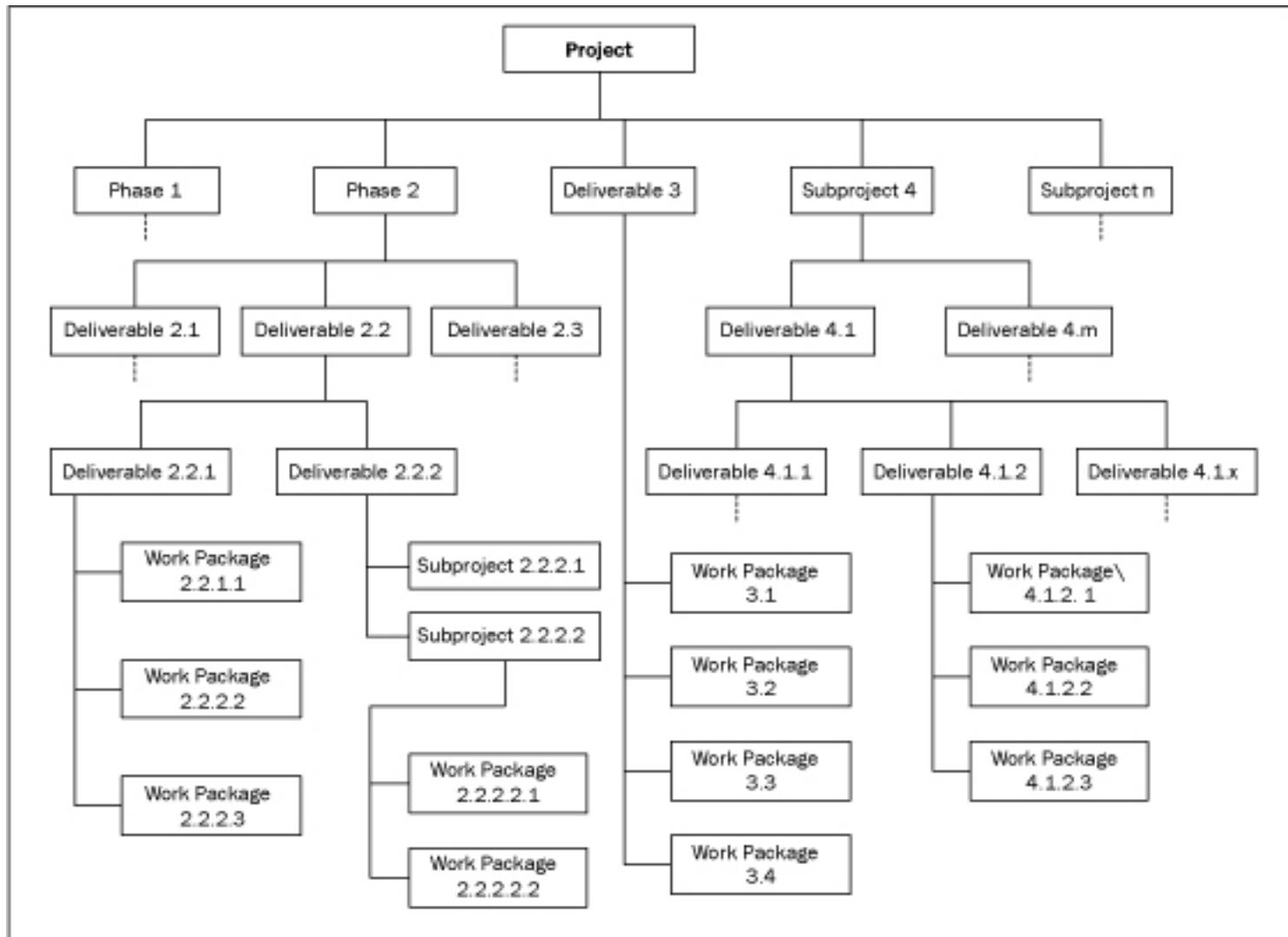
Recap: WBS



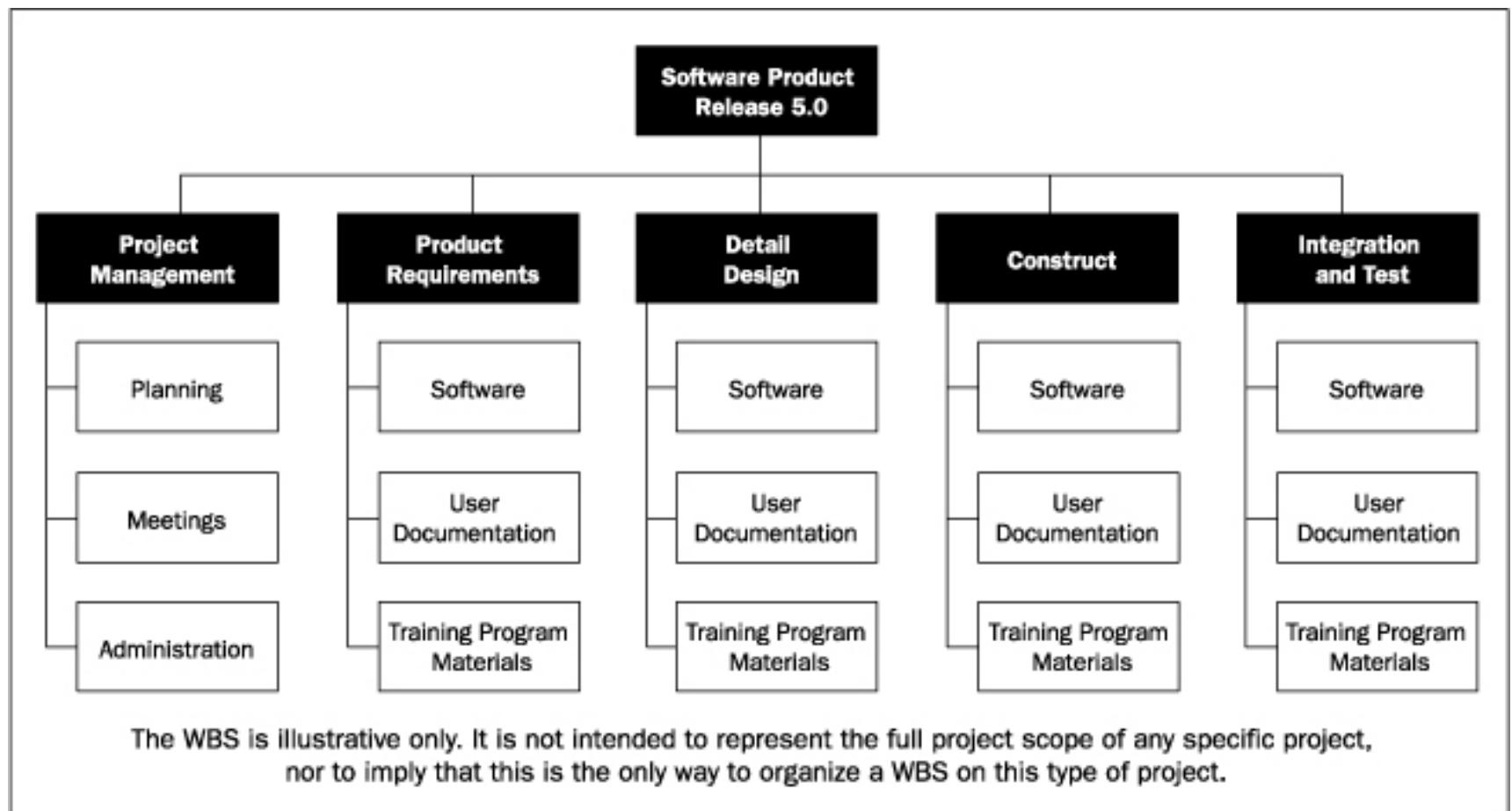
Recap: WBS RACI

	Responsible	Accountable	Consulted	Informed
Project Manager	X	X		
Project Team	X			
Project Review Authority				X
Project Stakeholders			X	X

Recap: WBS Structure by Phase



Recap: WBS By Discipline



The WBS is illustrative only. It is not intended to represent the full project scope of any specific project, nor to imply that this is the only way to organize a WBS on this type of project.

Topic –Time management

- Time Management
 - Estimation
 - Scheduling
- References
 - *Software Project Management: A Unified Framework*, Walker Royce, Addison Wesley
 - *Fundamentals of Project Management*, James P. Lewis, AMACOM Books
 - *Software Project Management*, Bob Huges, Mike Cotterrel
 - PMBOK

Planning the “WHEN”

- The distinction between project management and general management stays in scheduling
- ... but PM is not all about scheduling ☺

Time Management Process [PMBOK]

- Activity Definition
- Activity Sequencing
- Activity Resource Estimating
- Activity Duration Estimating
- Schedule Development
- Schedule Control

4 Activity Dependency Types

■ **Mandatory** Dependencies

- “Hard logic” dependencies
- Nature of the work dictates an ordering
- Ex: Coding has to precede testing
- Ex: UI design precedes UI implementation

■ **Discretionary** Dependencies

- “Soft logic” dependencies
- Determined by the project management team
- Process-driven
- Ex: Discretionary order of creating certain modules

4 Activity Dependency Types

■ **External** Dependencies

- Outside of the project itself
- Ex: Release of 3rd party product; contract signoff
- Ex: stakeholders, suppliers, year end

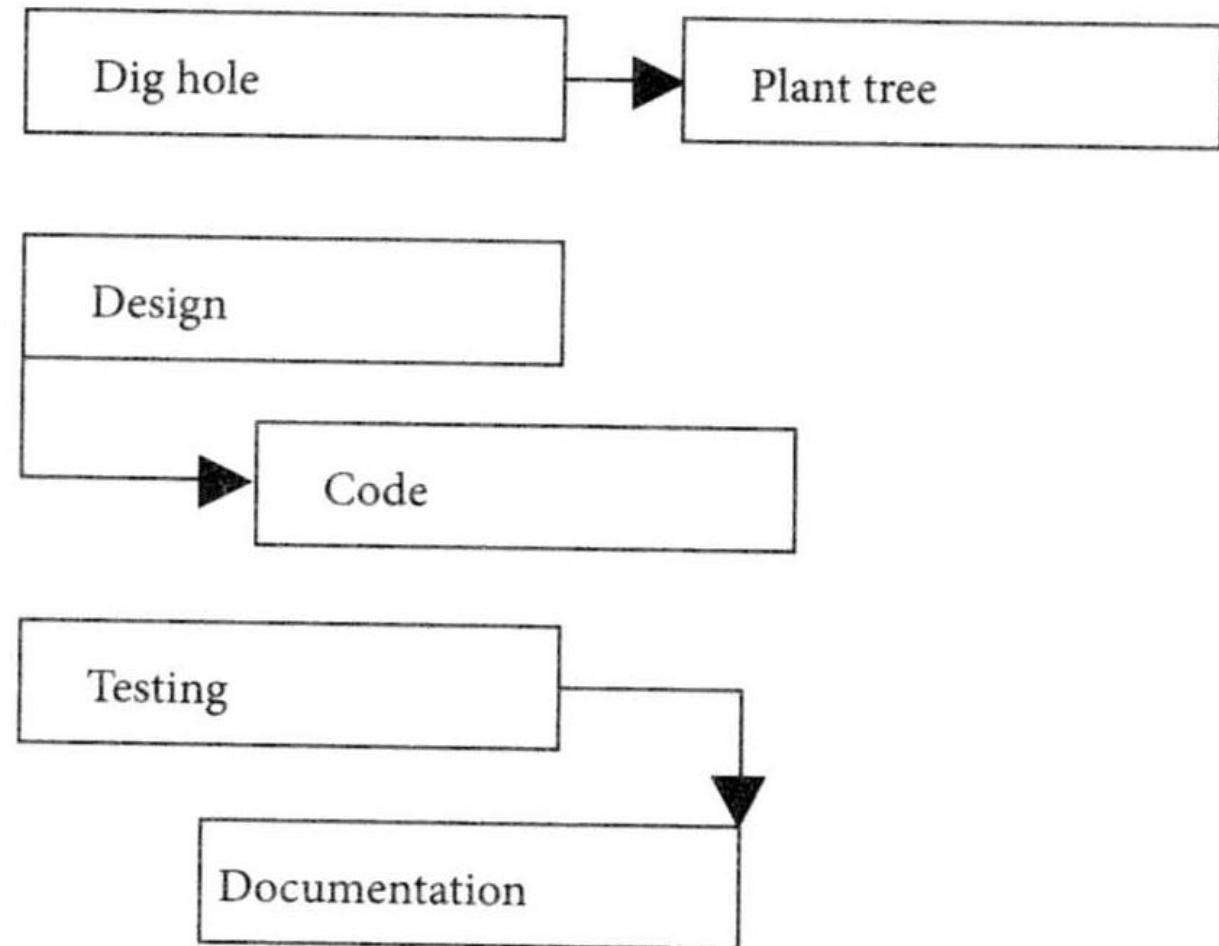
■ **Resource** Dependencies

- Two task rely on the same resource
- Ex: You have only one DBA but multiple DB tasks

Activity sequencing [1]

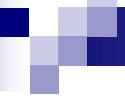
- Finish-to-start (FS) – the start of a task can be only after the finish of another
- Start-to-Start (SS) - start of one task triggers the start of another
- Finish-to-finish (FF) – finishing a task triggers the finish of another
- Start-to-Finish (SF) – start of one task triggers finish of another

Activity sequencing [2]



Constraints

- As Soon As Possible
- As Late As Possible
- Start No Earlier Than
- Finish No Earlier Than
- Start No Later Than
- Finish No Later Than
- Must Start On
- Must Finish On



Estimations

- Once tasks have been identified, the time and resources necessary to accomplish them must be determined.
- This process is called **estimating**.

Estimation

- **Effort/Work** = how much work will the activity need to be completed
- **Resources** = how many resources will be working on the activity
- **Duration** = how long will the activity last for

=> Estimate effort first

Effort

- Your best shot for providing estimations (how complex/how much work does the activity require?)
- Measured in man/month
 - Ex. 3 m-m = 1 person working for 3 months;
3 people working for one 1 month,
6 people working half-time for 1 month..
- Communication increases the time to complete activities!

Duration

- Measured in (work-)hours, (work-)days, (work-)months, ...
- Calendar time != duration: calendar time includes non-working days, holidays, ...
- Usually:
 - A duration of 5 days = 40 hours (8 hours a day) = 1 calendar week (Sat and Sun rest time)

A (simplistic) approach

- Duration = Effort/Resources
- Linear simplification (the non-linearity can be embedded in Effort)

Simplistic approach [2]

- When working with planning tools, you change one variable at a time.
- Standard characterization:
 - **Fixed Unit.** A task in which the **assigned resources** is a fixed value and any changes to the amount of work or the tasks duration do not affect the task units.
 - **Fixed Work.** A task in which the **amount of work** is a fixed value and any changes to the tasks duration or the number of assigned resources do not affect the tasks work.
 - **Fixed Duration.** A task in which the **duration** is a fixed value and any changes to the work or the assigned resources, don't affect the tasks duration.

Estimation approaches

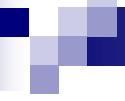
- To achieve credible results cost and schedule budgets should be done in two ways
 - Estimating **top-down**, starting from project level
 - Estimating **bottom-up** starting from task micro-analysis
 - **Parametric** estimating – use historical data adjusted to current project

Best practices

- Estimate based on the required level of accuracy
 - Rough estimation first (+/- 50-75%)
 - Second-round refinements (+/- 25%)
- Estimate more accurately as you progress

Scheduling

- Once tasks (from the WBS) and effort (from estimation) are known: then schedule
- **Primary** objectives
 - Best time
 - Least cost
 - Least risk
- **Secondary** objectives
 - Evaluation of schedule alternatives
 - Effective use of resources
 - Communications



Rules of thumb

- Organize tasks **concurrently** to make optimal use of workforce.
- **Minimize task dependencies** to avoid delays caused by one task waiting for another to complete.
- Dependent on project managers **intuition and experience**.

Terminology

■ Precedence:

- A task that must occur before another is said to have precedence of the other

■ Concurrence:

- Concurrent tasks are those that can occur at the same time (in parallel)

■ Leads & Lag Time

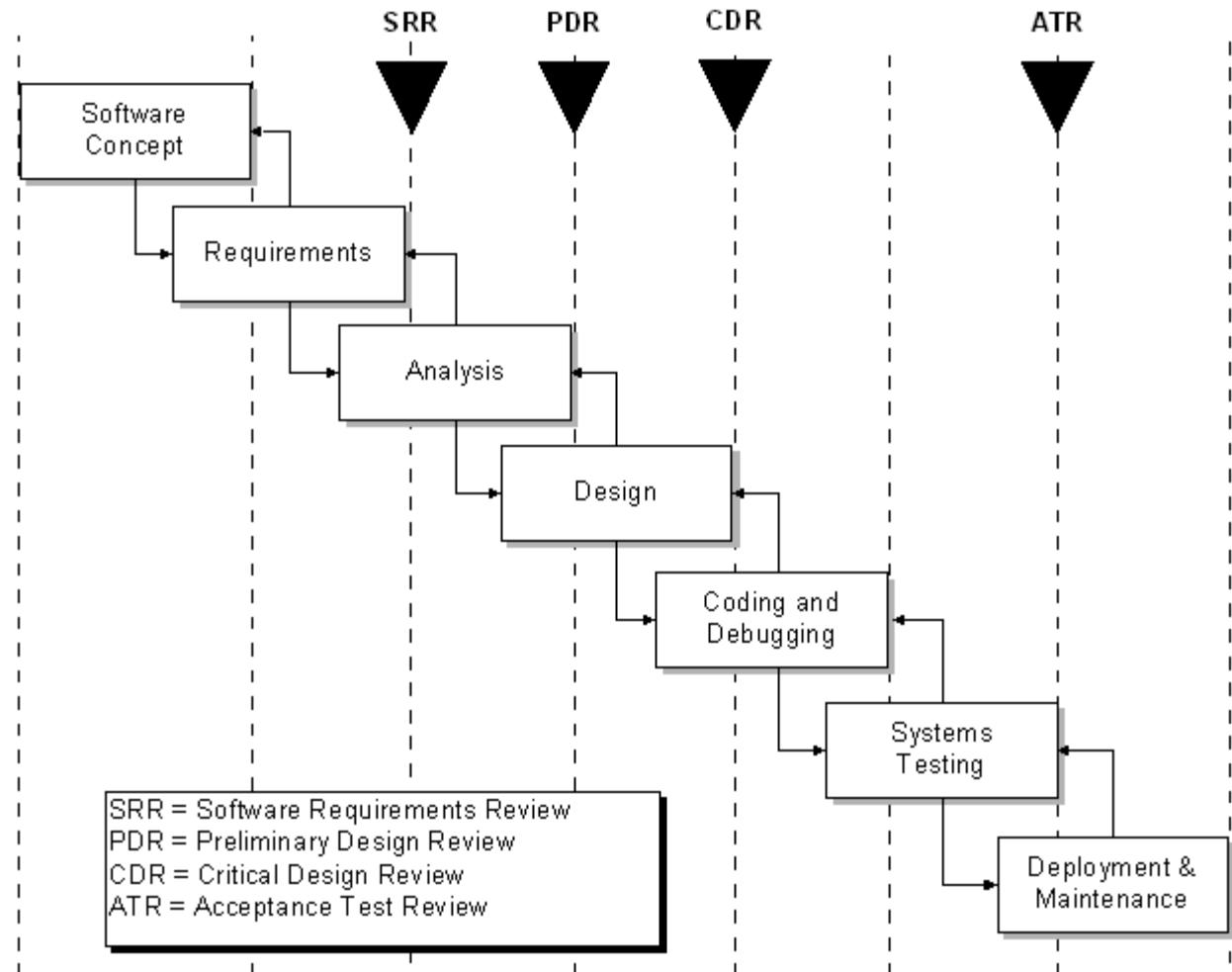
- Acceleration/Delay between activities

Terminology

- Milestones
 - Have a duration of zero
 - Identify critical points in your schedule
 - Often used at “review” or “delivery” times
 - at end or beginning of phases
 - Ex: Software Requirements Review (SRR)
 - Ex: User Sign-off
 - Can be tied to contract terms

Terminology

Example Milestones



Terminology

- Deliverable:
 - **a deliverable is a measurable and verifiable work product**
- ... in current practice sometimes milestone and deliverable are used interchangeably (both used to identify products - milestones may represent key-products)

Terminology

- Slack & Float
 - Synonymous terms
 - Free Slack
 - Slack an activity has before it delays next task
 - Total Slack
 - Slack an activity has before delaying whole project
 - Slack Time $T_S = T_L - T_E$
 - T_E = earliest time an event can take place
 - T_L = latest date it can occur w/o extending project's completion date

Schedule estimating

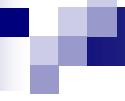
- Project Manager (and the management team) develop a characterization of overall size, process, environment, people and quality for the overall project
- A macro-level estimation of the total effort and schedule is developed
- A high-level WBS is developed
- Team leaders and developers estimate individual tasks

Schedule consolidation

- The lowest level WBS elements are defined into detailed tasks and schedule and budget is estimated by their responsible
- Estimates are consolidated and homogenized
- Comparison and validations are made

Padding estimates

- Bloated estimates – everyone adds some padding => totally inaccurate estimates
- Padding games - managers trim every estimate => more padding next time
- Lack of feedback – with no feedback no estimator can improve



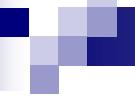
Safety margins

- Contingency funds and hours – depends on the risks list
 - 50-75% if high-risk project
 - 5-10% if low-risk project

- Management reserve

Early estimation method

- Use Case points estimations
 - UC Points Template.xls



■ Agile estimations

Planning game

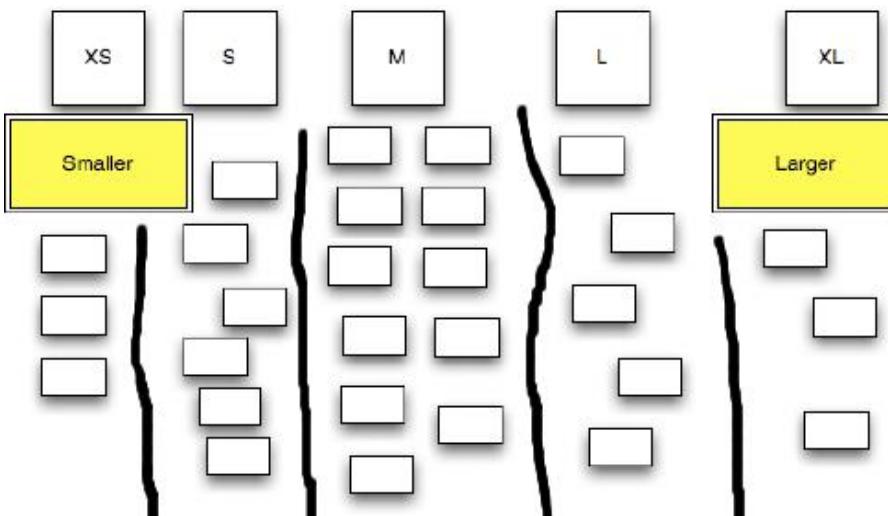
- Is based on user stories that reflect the features that should be included in the system.
- The project team reads and discusses the stories and ranks them in order of the amount of time they think it will take to implement the story.
- Stories are assigned '**story points**' reflecting their size and difficulty of implementation
- The number of effort points implemented per day is measured giving an estimate of the team's '**velocity**'
- This allows the total effort required to implement the system to be estimated

Story points

- Relative measurements (to pieces of work)
- Unit-less
- Build on user stories
- Unit = 1 story point = the effort needed to build a simple (or easy to estimate) product requirement or user story (from the backlog)
- Advantages
 - Relative estimates are more accurate
- Limitations
 - Conceptual shift from time -> points (don't confuse hours with points)
 - Tendency to inflate the story points to give appearance of productivity

Points examples

- Planning poker
- Affinity grouping



Velocity estimates

- **Velocity** is a measurement of the work that can be taken on and finished over the length of a sprint
- Velocity = #story points/time
- Helps
 - Estimating the effort the development team can achieve throughout the project
 - Estimating effort that can be achieved on future work
 - Indirectly tracking the team's productivity
- USE THE SAME MEASUREMENT
- Story points HAVE TO BE COMPLETED! (Definition of “done”)

Velocity-driven development

- Velocity of previous sprints is used as a basis for planning other sprints in the project
- Requires a certain amount of **stability** in past velocity (velocities become more consistent over the course of a project)
- Using **previous data** to determine estimates:
 - **Linear extrapolation** (extending graphed data beyond the known data in a straight line based on how previous data has behaved),
 - **Averaging** the velocity of the three previous sprints,
 - Using the **minimum** velocity of the three previous sprints.

Scheduling Techniques

■ Network Diagrams

- Critical Path Method (CPM)
- PERT

■ Bar Charts

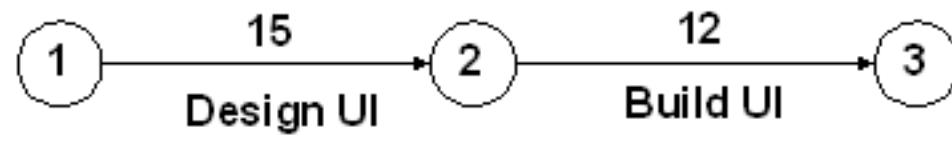
- Milestone Chart
- Gantt Chart

Network Diagrams

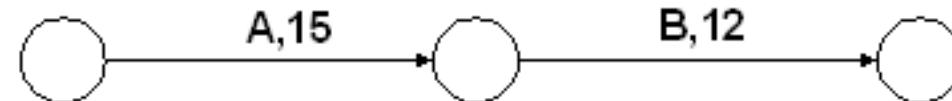
- Two classic formats
 - AOA: Activity on Arrow
 - AON: Activity on Node
- Each task labeled with
 - Identifier (usually a letter/code)
 - Duration (in std. unit like days)
- There are other variations of labeling
- There is 1 start & 1 end event
- Time goes from left to right

Node Formats

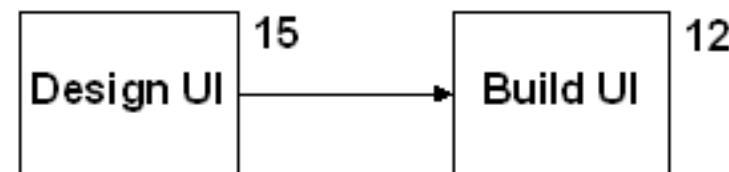
Activity on Arrow (AOA)



or



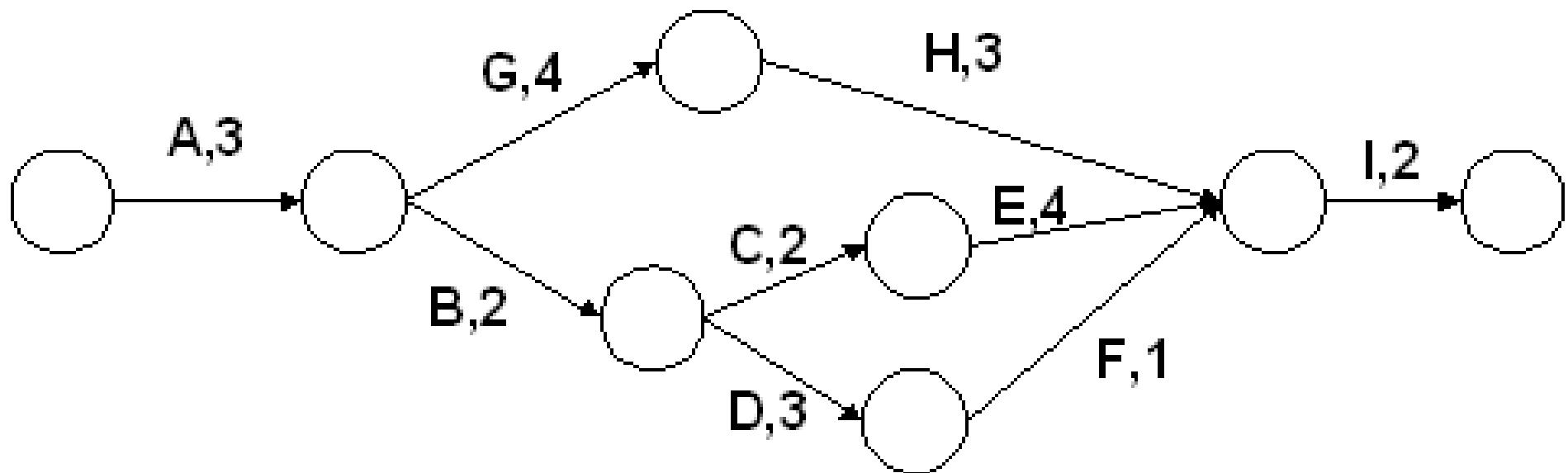
Activity on Node (AON)



or

Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

Critical Path Example



$$AGHI = 3 + 4 + 3 + 2 = 12$$

$$ABCDEI = 3 + 2 + 2 + 4 + 2 = 13$$

$$ABDFI = 3 + 2 + 3 + 1 + 2 = 11$$

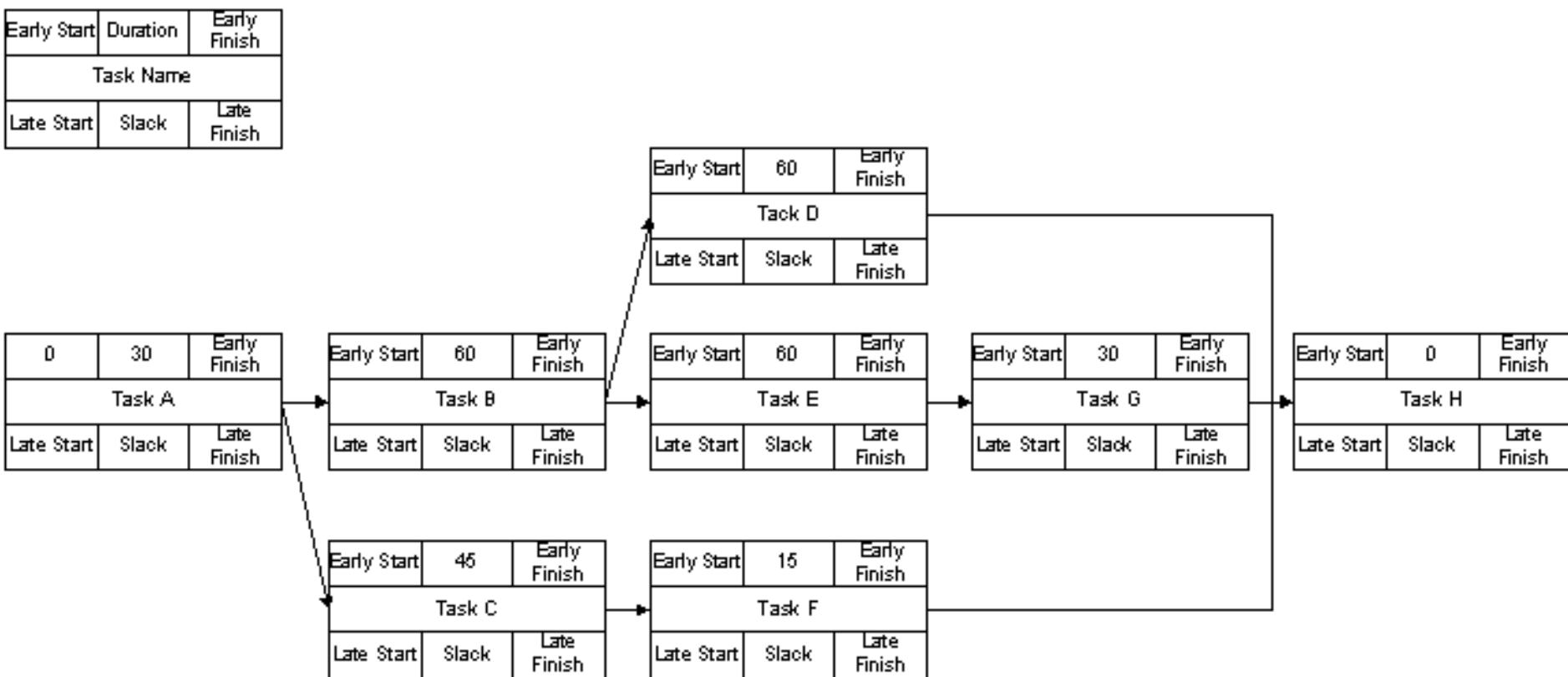
Critical Path

- the sequence of project network activities which add up to the **longest overall duration**.
- => the shortest time possible to complete the project

CPM

- Critical Path Method
 - The process for determining and optimizing the critical path
- Non-CP tasks can start earlier or later w/o impacting completion date
- Note: Critical Path may change to another as you shorten the current

Example Step 1



Forward Pass

- To determine early start (ES) and early finish (EF) times for each task
- Work from left to right
- Adding times in each path
- Rule: when several tasks converge, the ES for the next task is the largest of preceding EF times

Example Step 2

Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

0	30	30
Task A		
Late Start	Slack	Late Finish

30	60	90
Task B		
Late Start	Slack	Late Finish

90	60	150
Task D		
Late Start	Slack	Late Finish

90	60	150
Task E		
Late Start	Slack	Late Finish

150	30	180
Task G		
Late Start	Slack	Late Finish

180	0	180
Task H		
Late Start	Slack	Late Finish

30	45	75
Task C		
Late Start	Slack	Late Finish

75	15	90
Task F		
Late Start	Slack	Late Finish

Backward Pass

- To determine the last finish (LF) and last start (LS) times
- Start at the end node
- Compute the bottom pair of numbers
- Subtract duration from connecting node's earliest start time

Example Step 3

Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

0	30	30
Task A		
Late Start	Slack	Late Finish

30	60	90
Task B		
Late Start	Slack	Late Finish

30	45	75
Task C		
Late Start	Slack	Late Finish

90	60	150
Task D		
120	Slack	180

90	60	150
Task E		
90	Slack	150

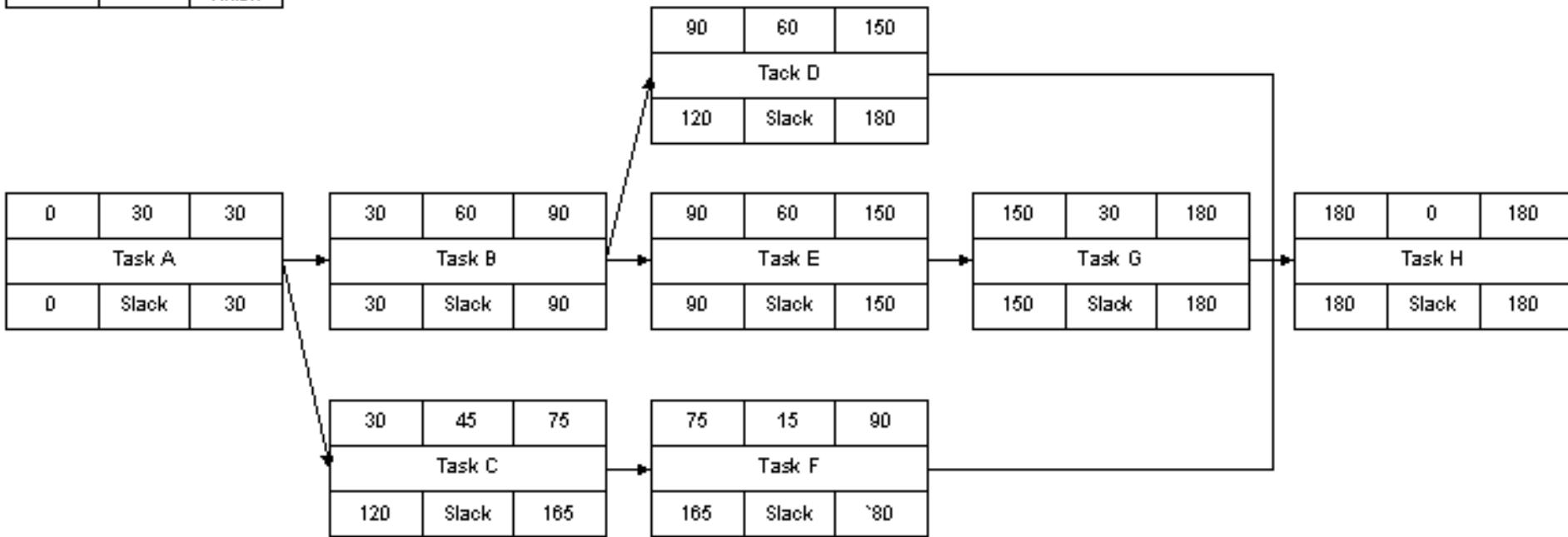
75	15	90
Task F		
Late Start	Slack	Late Finish

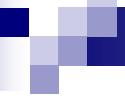
150	30	180
Task G		
150	Slack	180

180	0	180
Task H		
180	Slack	180

Example Step 4

Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

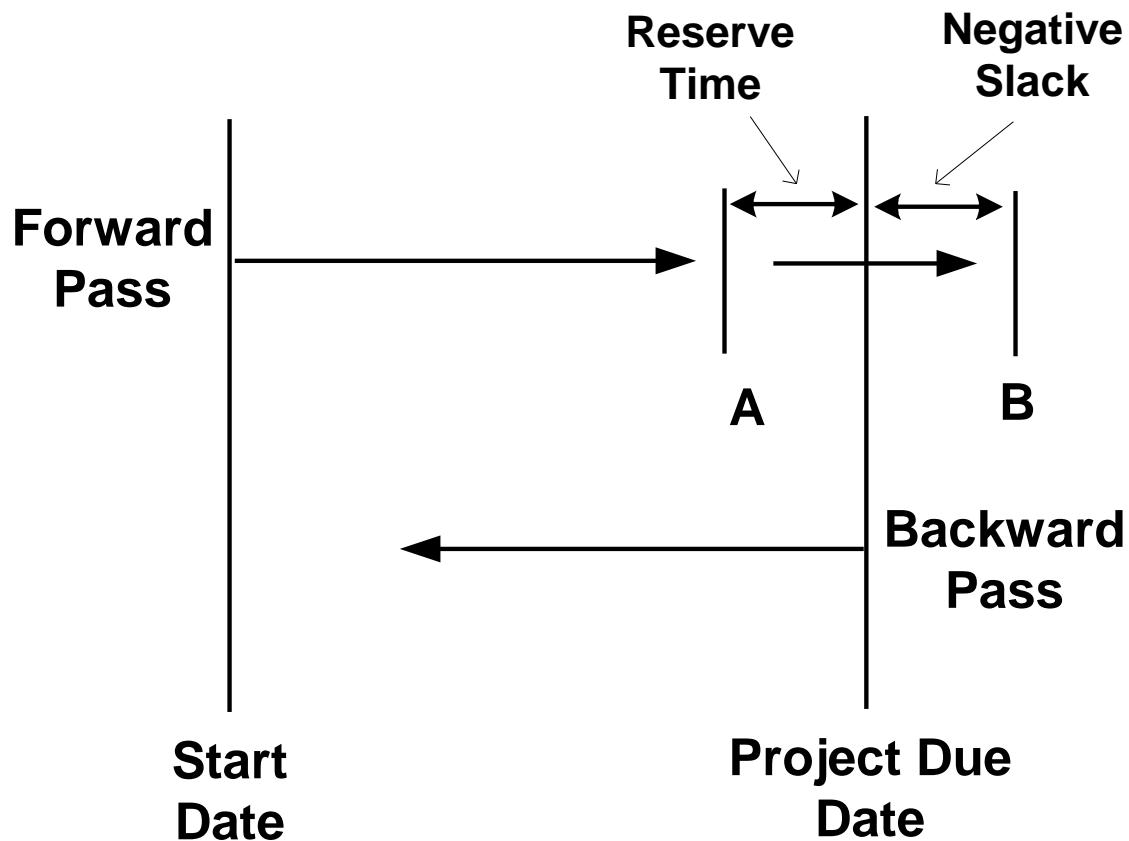




Slack & Reserve

- Can slack be negative?
- What does that mean?
- How can you address that situation?

Slack & Reserve



Network Diagrams

- Advantages
 - Show precedence well
 - Reveal interdependencies not shown in other techniques
 - Ability to calculate critical path
 - Ability to perform “what if” exercises
- Disadvantages
 - Default model assumes resources are unlimited
 - You need to incorporate this yourself (Resource Dependencies) when determining the “real” Critical Path
 - Difficult to follow on large projects

Scheduling with uncertain durations

- PERT
 - apply Critical Path method and analyze from a probabilistic perspective
- Monte Carlo Simulations
 - calculates sets of artificial (but realistic) activity duration times and then applies a deterministic scheduling procedure to each set of durations

PERT

- Program Evaluation and Review Technique
- Based on idea that estimates are uncertain
 - Therefore uses duration **ranges**
 - And the **probability** of falling to a given range
- Uses an “expected value” (or weighted average) to determine durations
- Use the following methods to calculate the expected durations, then use as input to your network diagram

PERT Formula

- estimate a task duration: weighted average

$$t_e = \frac{a+4m+b}{6}$$

where t_e – expected time

a – optimistic estimate (5% chance that duration $< a$)

m – most likely estimate (modal value of the distribution)

b – pessimistic estimate (5% chance that duration $> b$)

PERT Formula

- Confidence Interval can be determined
- Based on a standard deviation of the expected time
 - Using a bell curve (normal distribution)

$$s = \frac{b - a}{6}$$

- For the whole critical path use

$$s_{cp} = \sqrt{s_1^2 + s_2^2 + \dots + s_n^2}$$

PERT Example

Description	Planner 1	Planner 2
m	10d	10d
a	9d	9d
b	12d	20d
PERT time	10.16d	11.5d
Std. Dev.	0.5d	1.8d

- Confidence interval for P2 is 4 times wider than P1 for a given probability
- Ex: 68% probability of 9.7 to 11.7 days (P1) vs. 9.5-13.5 days (P2)

PERT

- Advantages
 - Accounts for uncertainty
- Disadvantages
 - Focuses on one critic path (that might vary during the project)
 - Does not consider correlations between activities
 - Needs 3 estimations for each activity (increased effort)
- Get PERT software to calculate it for you

CPM vs. PERT

- Both use Network Diagrams
- CPM: deterministic
- PERT: probabilistic
- CPM: one estimate, PERT: three estimates
- PERT is infrequently used

Monte Carlo Simulation Example

3 activities in series, normal distribution of durations

Activity	Mean (Days)	Standard Deviation (Days)
A	2.5	1.5
B	5.6	2.4
C	2.4	2.0

Mean duration: $2.5+5.6+2.4 = 10.5$

$SD = \sqrt{1.5^2 + 2.4^2 + 2^2} = 3.5$

Simulation Number	Activity A	Activity B	Activity C	Project Duration
1	1.53	6.94	1.04	9.51
2	2.67	4.83	2.17	9.66
3	3.36	6.86	5.56	15.78
4	0.39	7.65	2.17	10.22
5	2.50	5.82	1.74	10.06
6	2.77	8.71	4.03	15.51
7	3.83	2.05	1.10	6.96
8	3.73	10.57	3.24	17.53
9	1.06	3.68	2.47	7.22
10	1.17	0.86	1.37	3.40
11	1.68	9.47	0.13	11.27
12	0.37	6.66	1.70	8.72

Estimated Mean Project Duration = 10.49

Estimated Standard Deviation of Project Duration = 4.06

Exercise

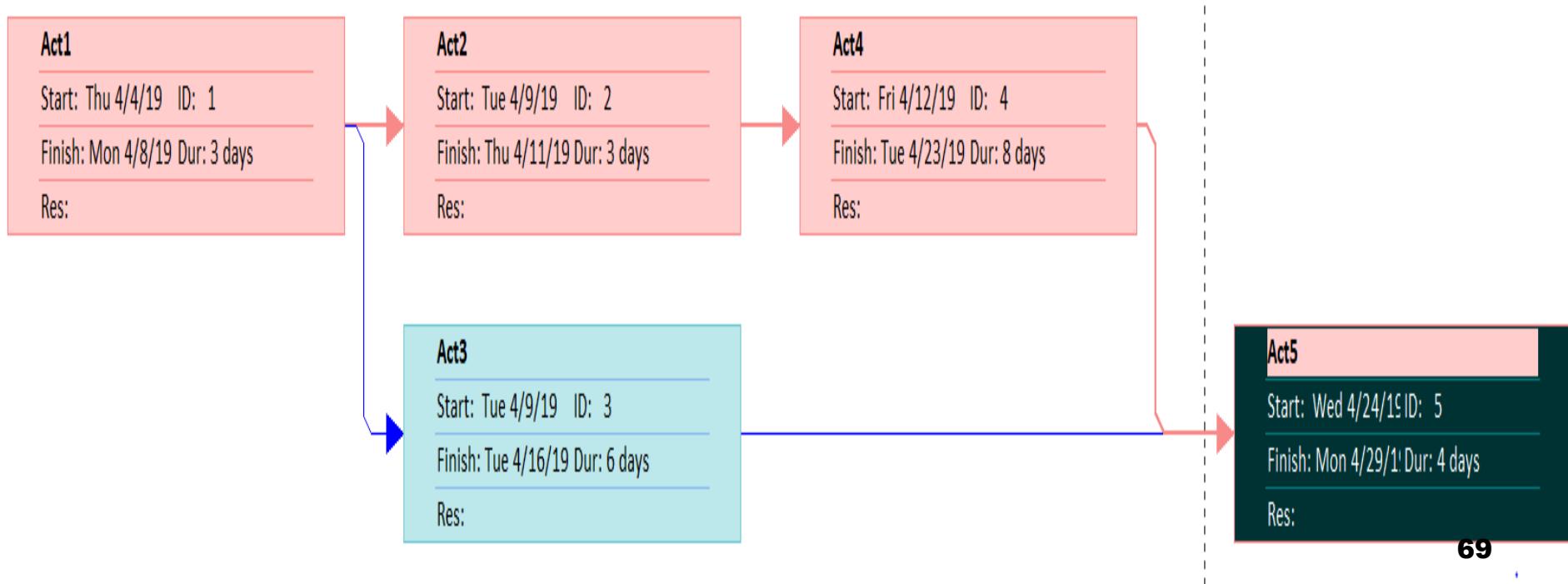
- Act1 starts immediately and is estimated to 3 days.
- Act2 can start after Act1 is completed and is estimated to 3 days.
- Act3 can start after Act1 is completed and is estimated to 6 days
- Act4 can start after Act2 is completed and is estimated to 8 days
- Act5 can start after Act4 and Act3 are completed and is estimated to 4 days

Path1 = 3+3+8+4 = 18

Path2 = 3+6+4 = 13

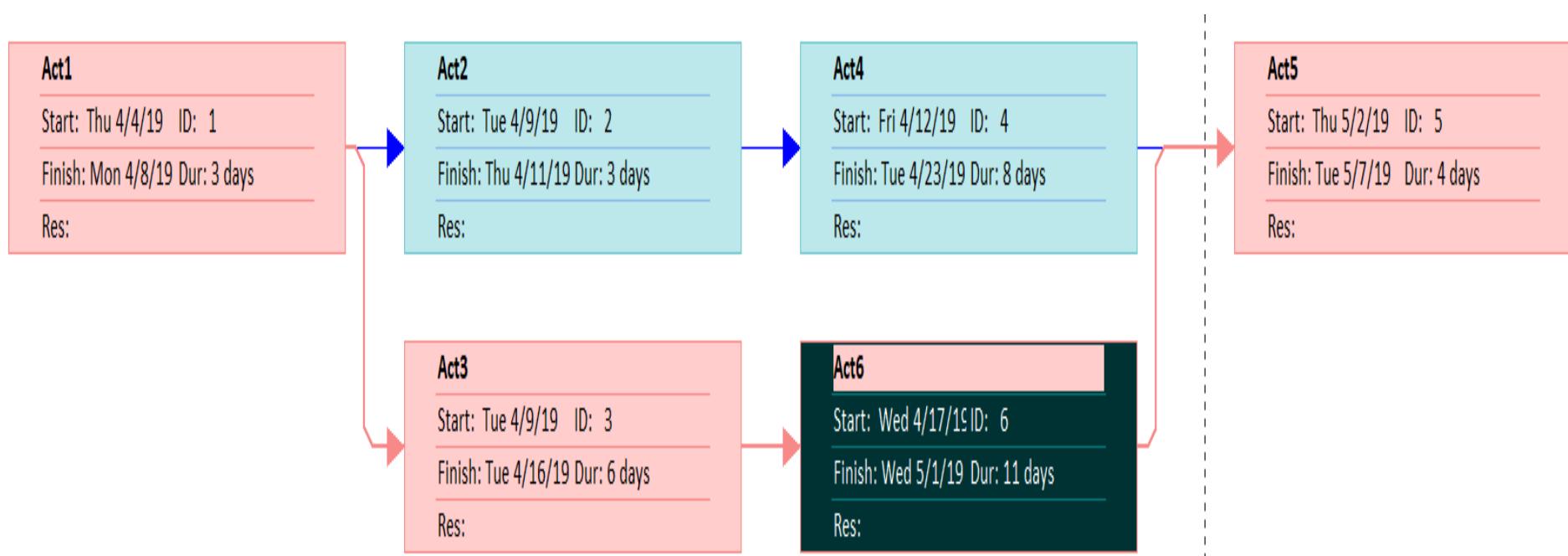
Slack for Act3?

5



- What if Act6 has to be added?
- It takes 11 days, should be done after Act3 and before Act5.
- New CP?

$$3+6+11+4 = 24$$



Wrap-up

- Time management
 - Estimation
 - Activity dependencies
 - Scheduling
- Techniques
 - CPM
 - PERT
 - Monte Carlo Simulations

Quiz time

- Let's switch over to Moodle

Next time discussion topic

- Starting from a list of activities, their estimations and their dependencies, build a network diagram and highlight the critical path(s).

Next time

- Resource allocation and schedule compression



Resource management

Lecture 7

...recap::WBS and Scheduling

■ Answer the questions

- “What must be done?”
- “When will it be done?”
- “Who will build it?”
- “How much will it cost?”

...recap::WBS and Scheduling

■ PMBOK

- WBS is part of Project Scope Management
- Scheduling is part of Time Management Process

...recap::WBS

- Definition:
 - A *WBS* is a *hierarchy of elements that decomposes the project plan into discrete work tasks.*
- WBS Elements
 - Delineation of all significant work
 - Clear task decomposition for assignment and responsibilities
 - Framework for scheduling, budgeting and expenditure tracking

...recap::WBS – The Project Plan Architecture

- WBS is the project's plan “architecture”
 - It must encapsulate change
 - It must evolve with the appropriate level of detail through the project lifecycle
 - It must cover ALL project tasks
- WBS Deliverables must be SMART:
Specific, Measurable, Attainable,
Relevant, Time-limited

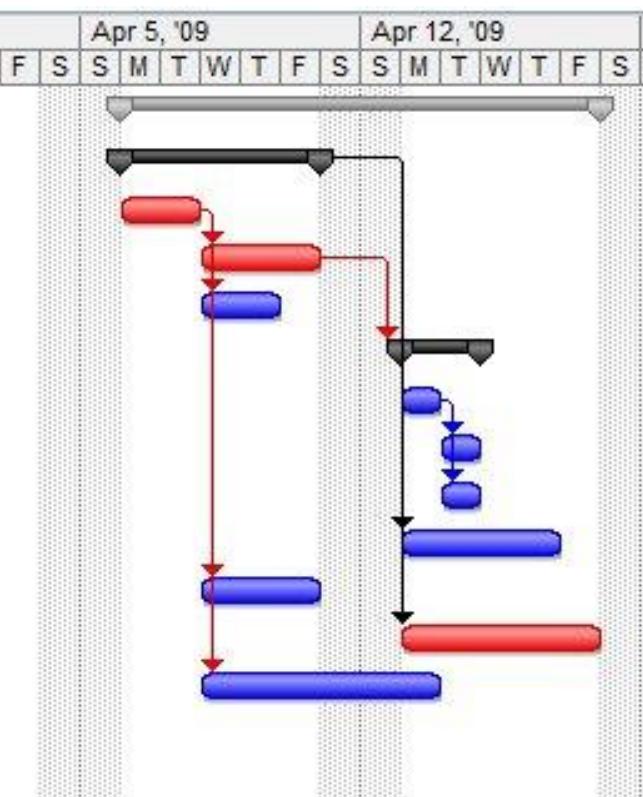
...recap::Project Schedule

- Estimating
 - Once tasks have been identified, the time and resources necessary to accomplish them must be determined.

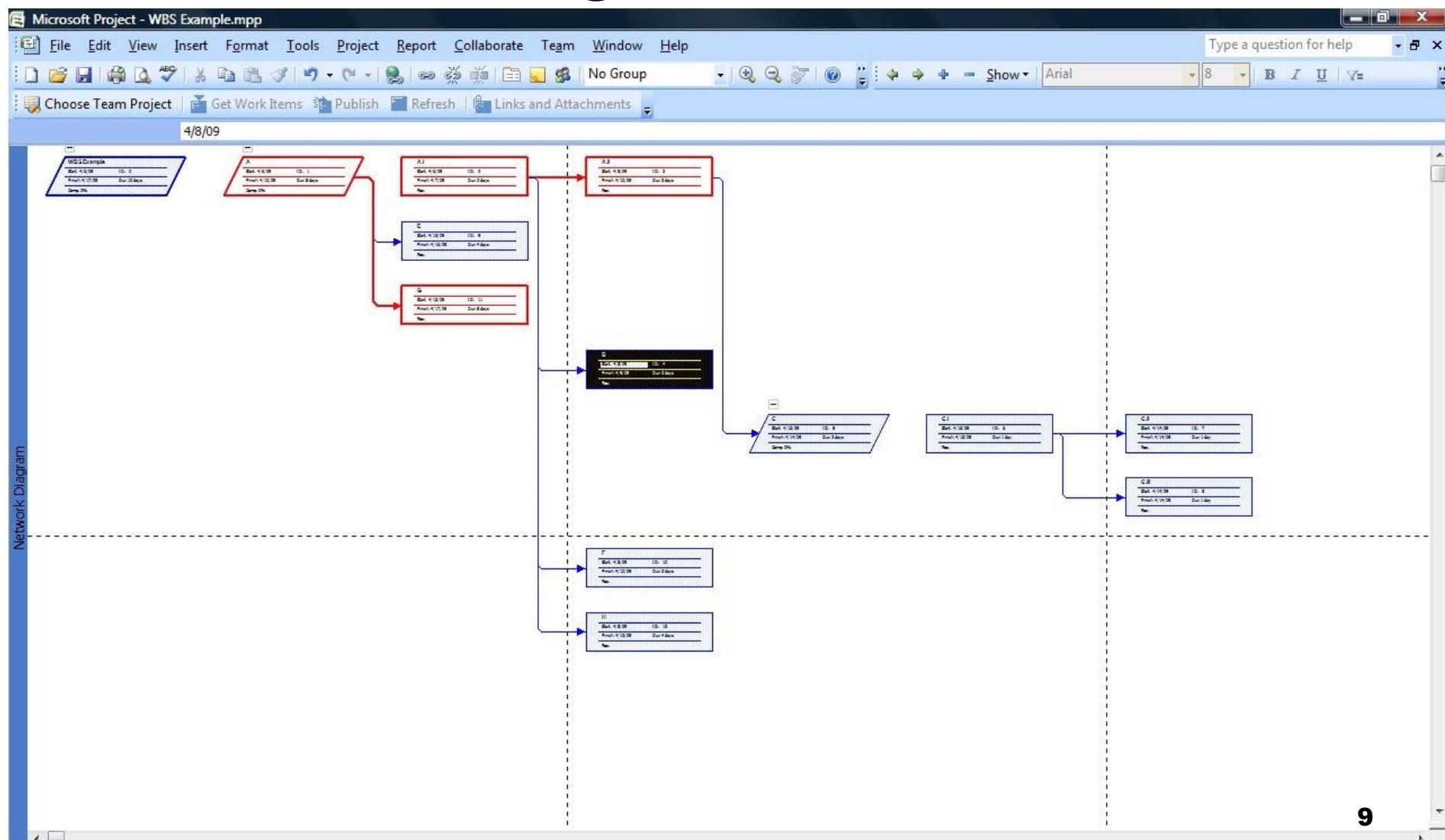
...recap::Schedule Estimating

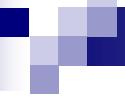
- Project Manager (and the management team) develop a characterization of overall size, process, environment, people and quality for the overall project
- A macro-level estimation of the total effort and schedule is developed
- A high-level WBS is developed
- Team leaders and developers estimate individual tasks

GANTT (MS Project)



Network Diagrams





Resource Management

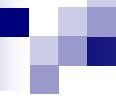
- Resource Allocation
- Schedule compression

Reference

- PMBOK
- Righting Software by Juval Löwy, 2020
Pearson Education (Chapters 7,8,9)

Project resources are about...

- **Capacity** – resources allow us to get more done than we can do alone
 - People provide extra minds and bodies to get things done
 - Tools and equipment amplify our efficiency
- **Capability** – resources allow us to do things we cannot otherwise do
 - People bring new skills, experience, and expertise
 - Tools and equipment allow us to do things people cannot



Challenges of resource management

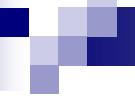
To get the

- Right resources...
- in the right quantities...
- to the right specifications, quality standards, or with the right capabilities...
- in the right place...
- at the right time.

PMBOK Resource Management

- “*Project Resource Management includes the processes to identify, acquire, and manage the resources needed for the successful completion of the project.*”

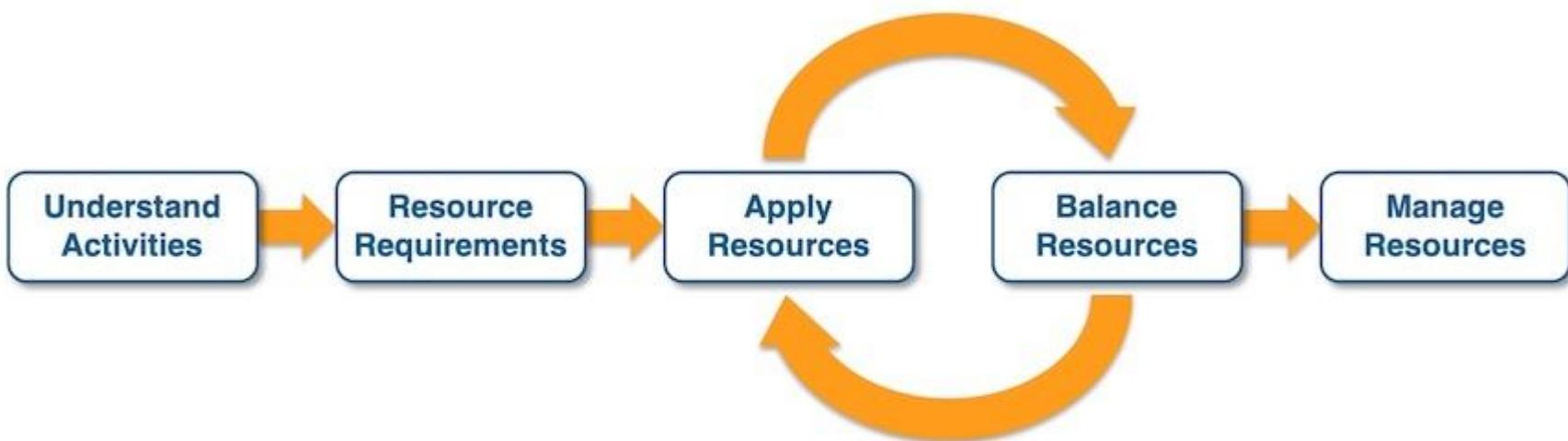
the Project Management Institute ([PMI](#)), in the PMBOK Guide 6th Edition



Resource management processes

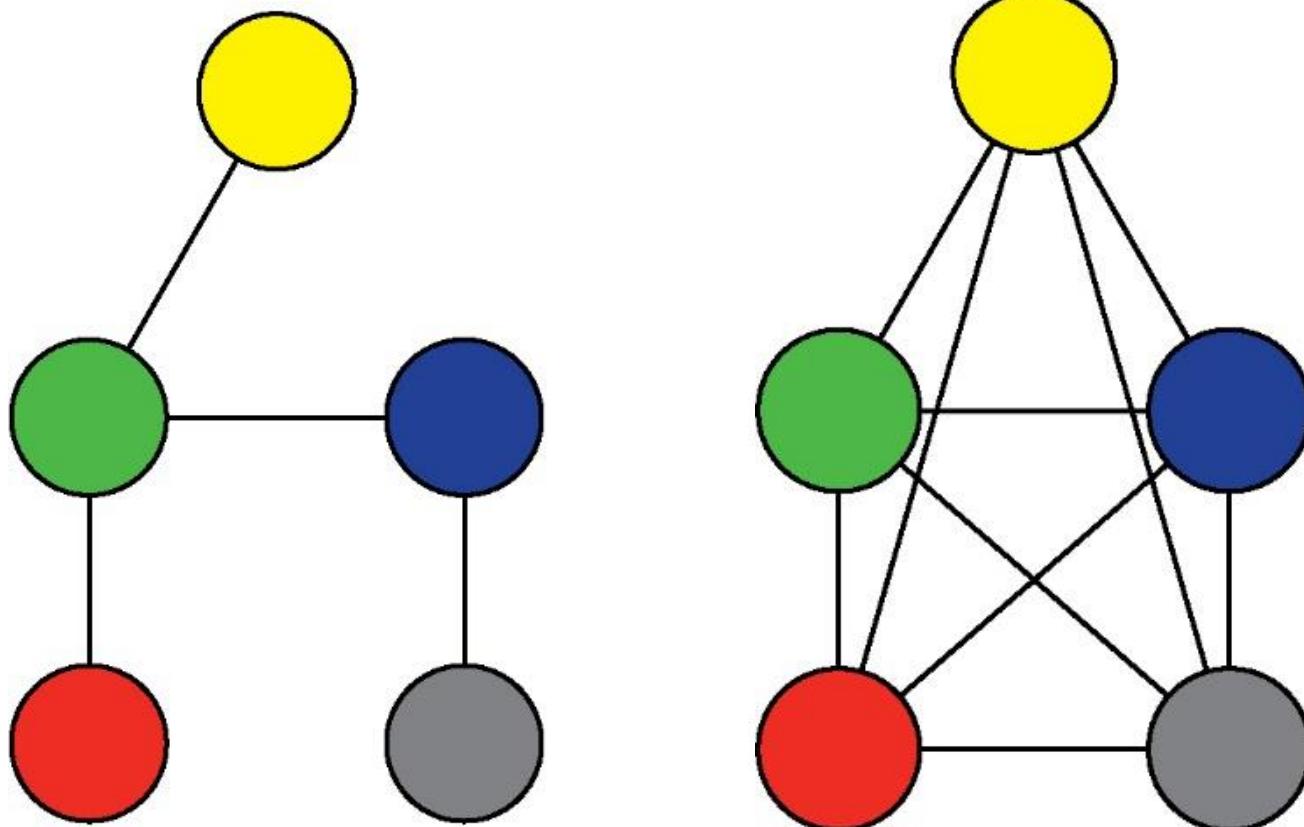
- Plan Resource Management
- Estimate Activity Resources
 - what resources you'll need and the quantities
- Acquire Resources
- Develop Team
- Manage Team
- Control Resources

Resource Planning



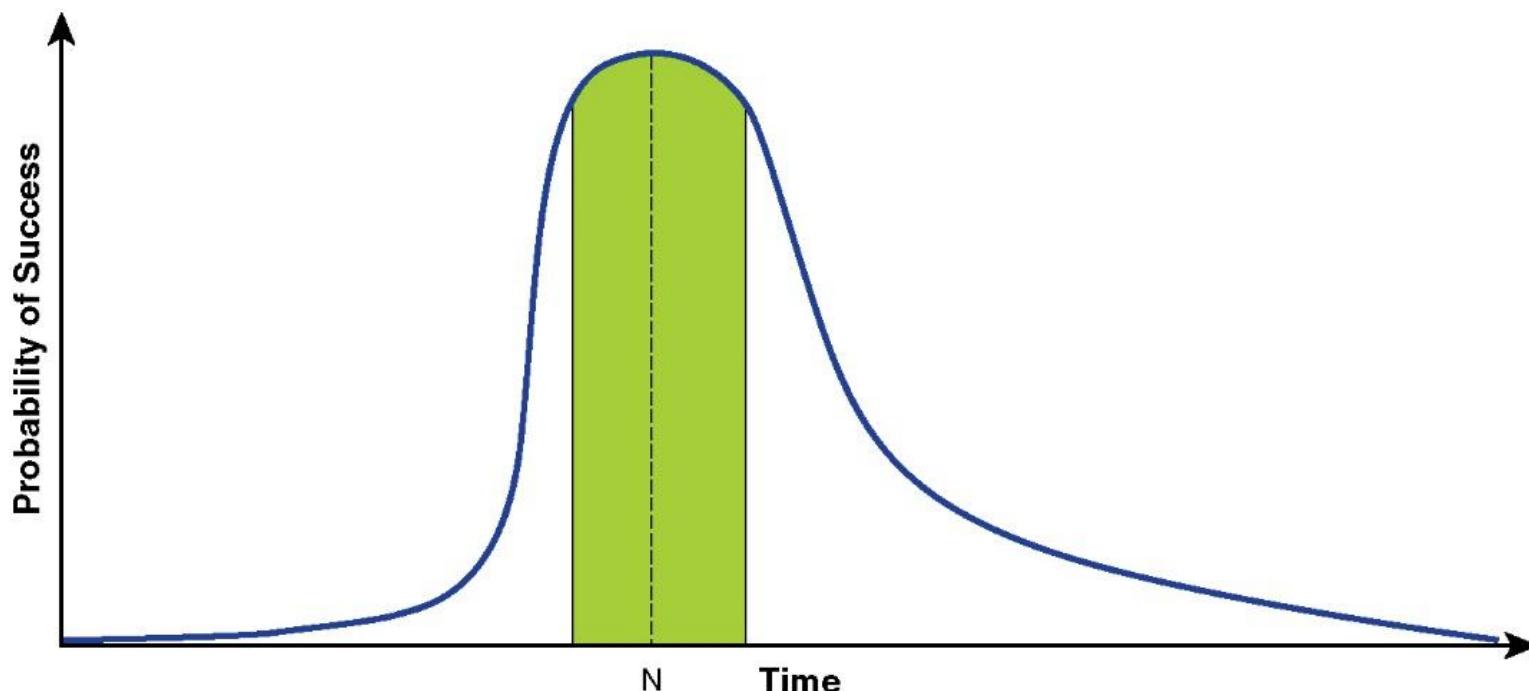
The Resource Planning Process

Understand the activities - WBS

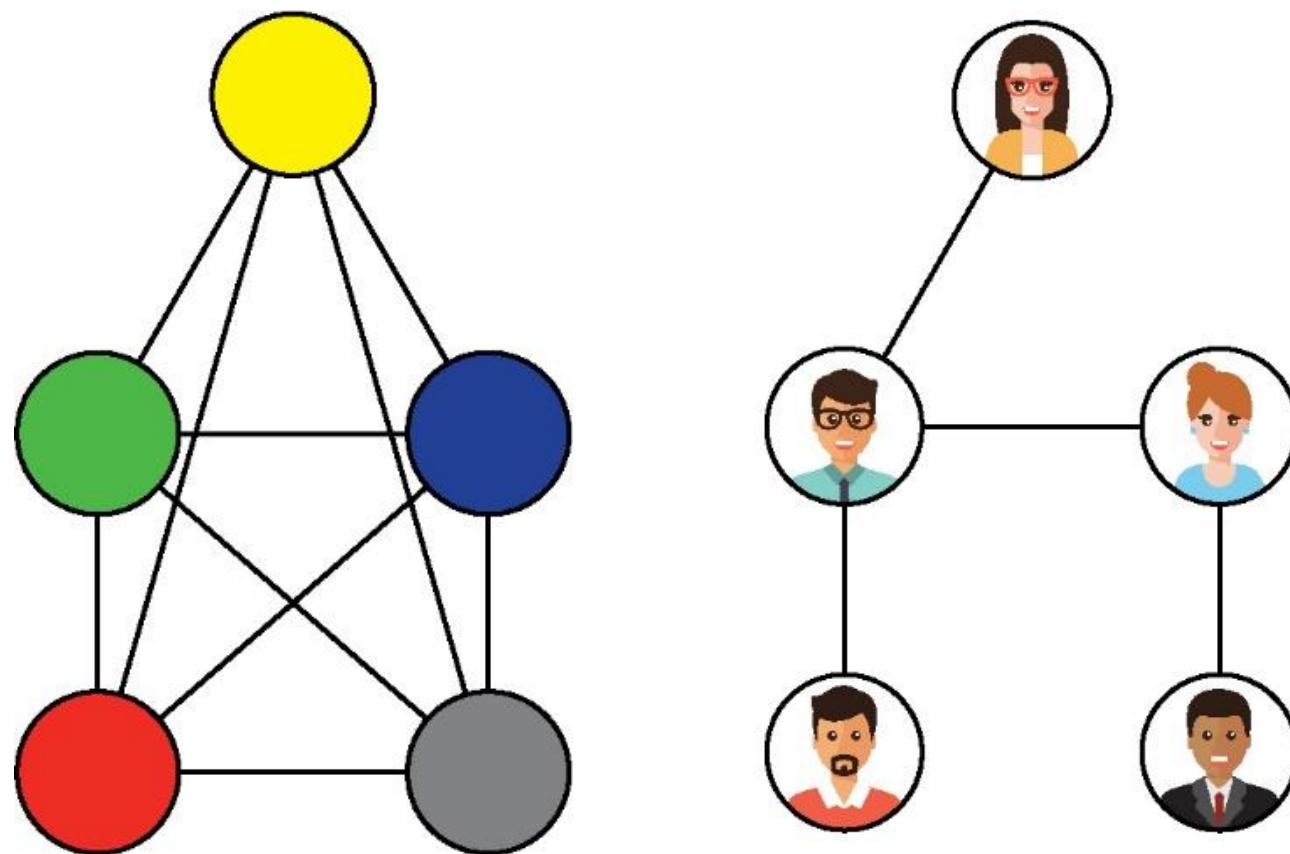


Resource requirements

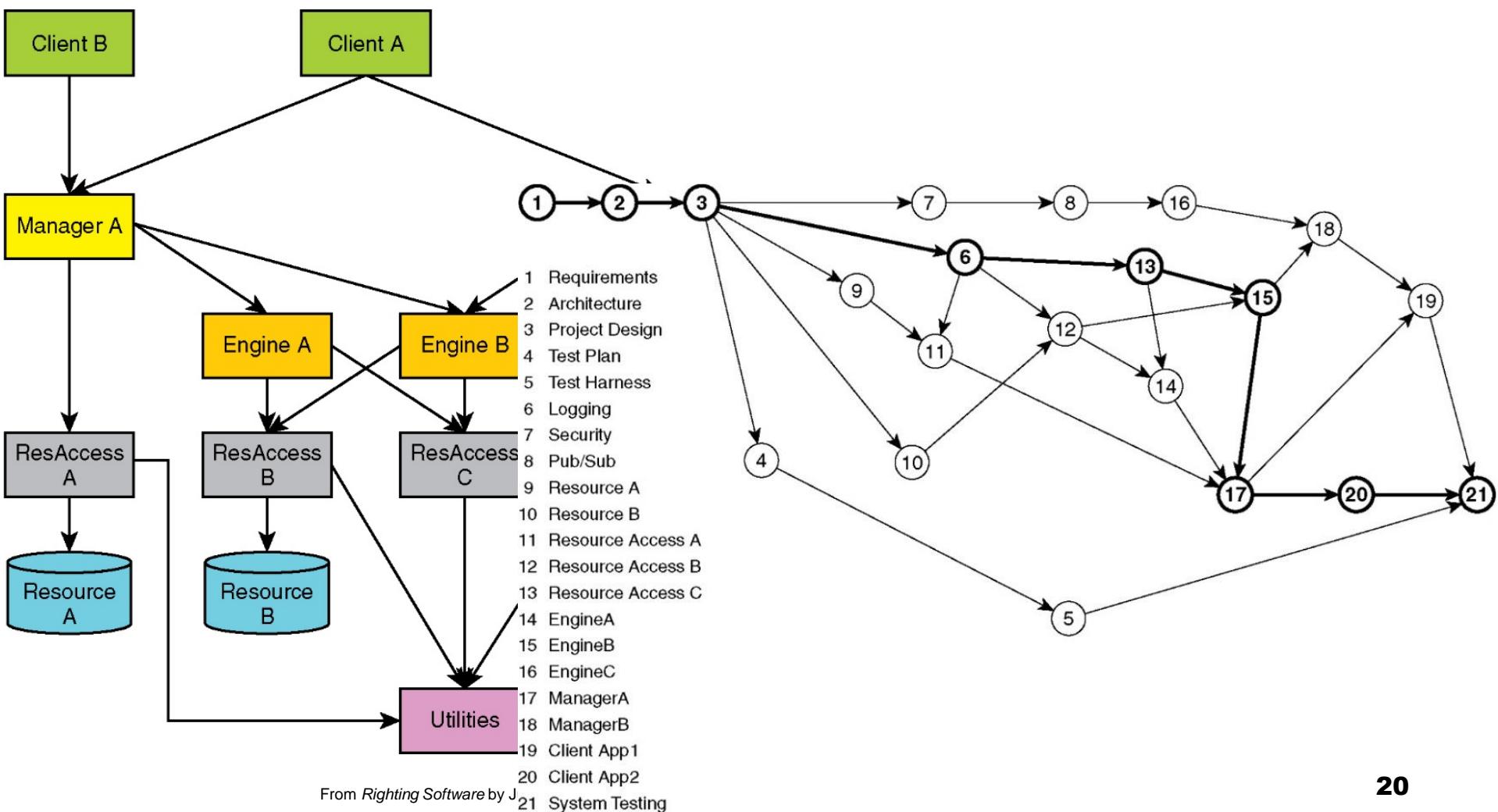
- Type
- Duration (estimations)



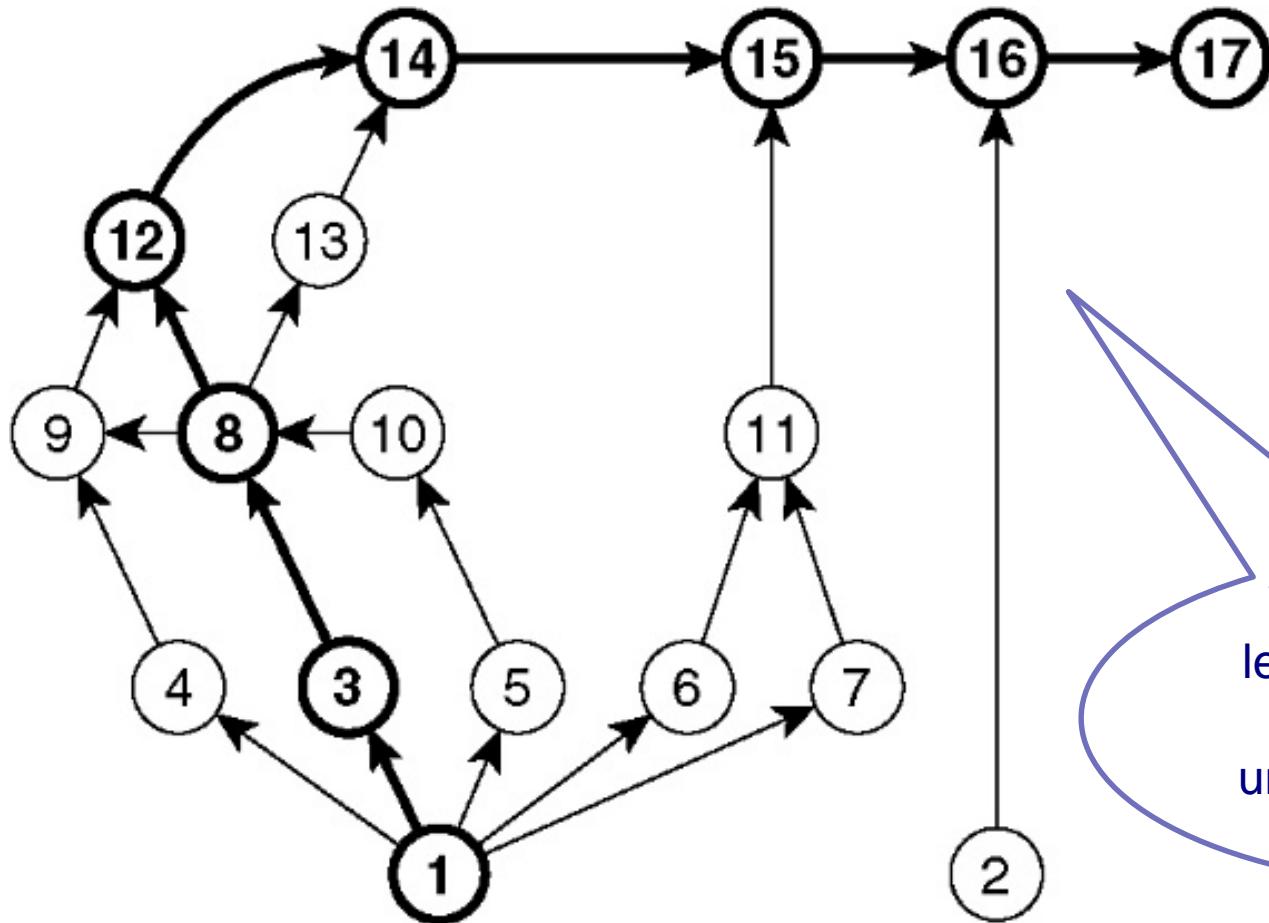
Apply resources to the activities



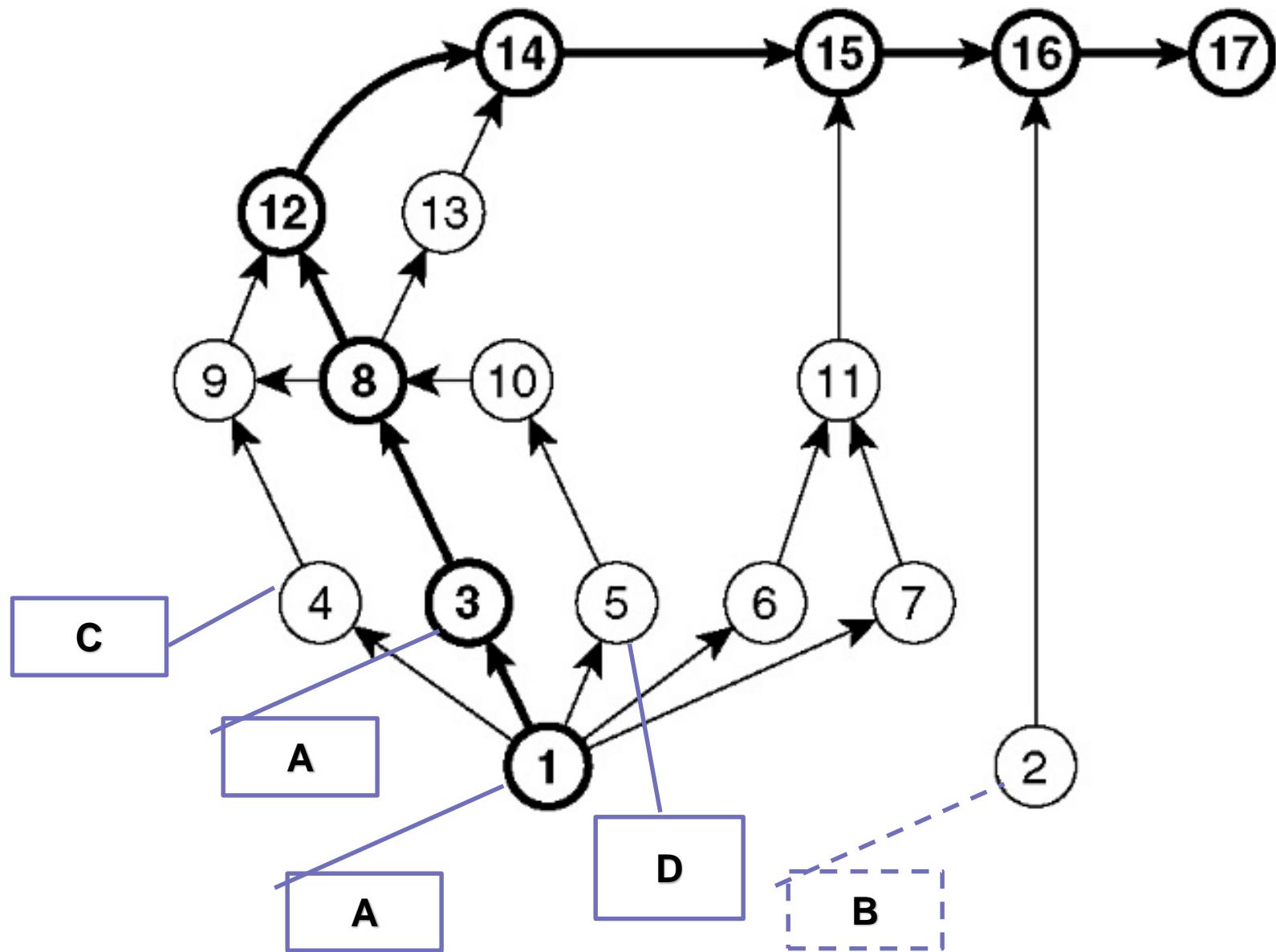
Architecture -> Project design



Discovering staffing levels



What is the lowest level of resources so we can progress unimpeded along the CP?

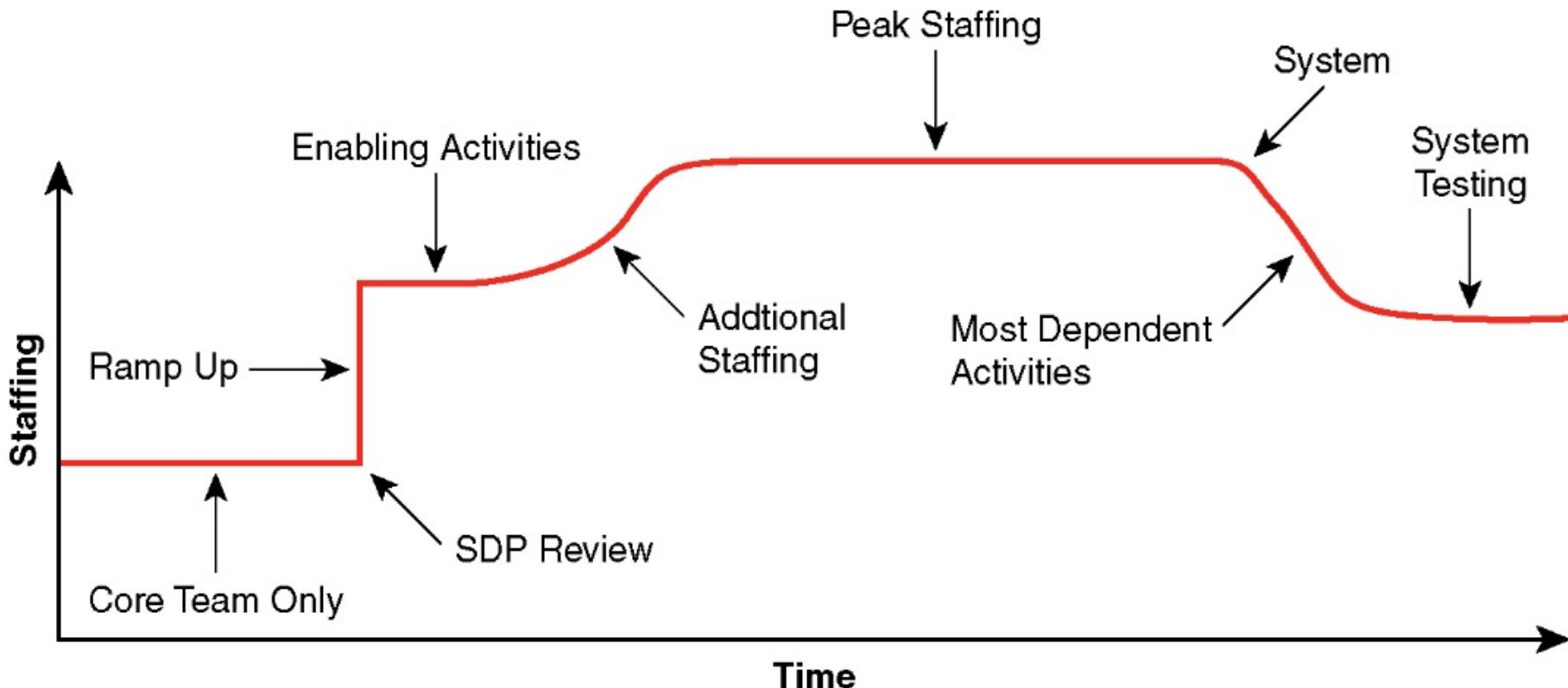


Iterative process

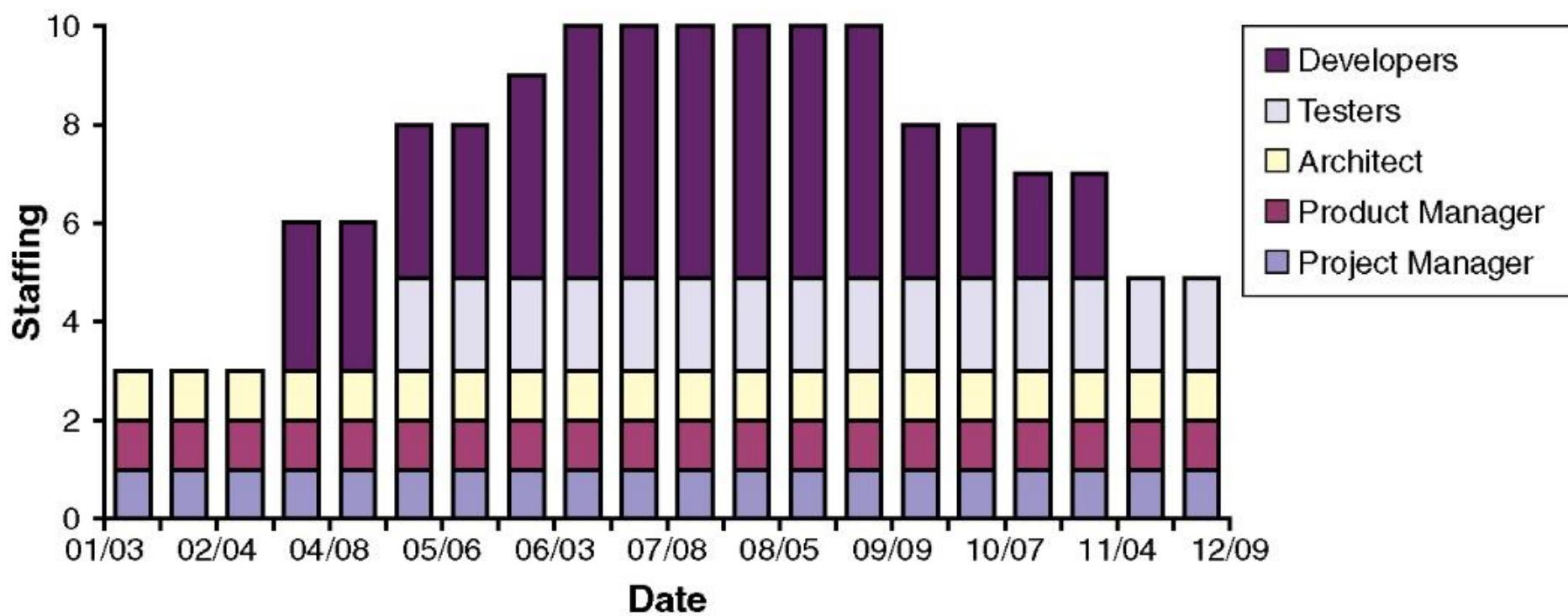
- Start with a slightly oversized resource level
- Assign resources based on the float (from low to high)
- If success => retry with fewer resources
- If non-critical activities become critical => project cannot be done so go back to previous set of resources

Correct staffing distribution

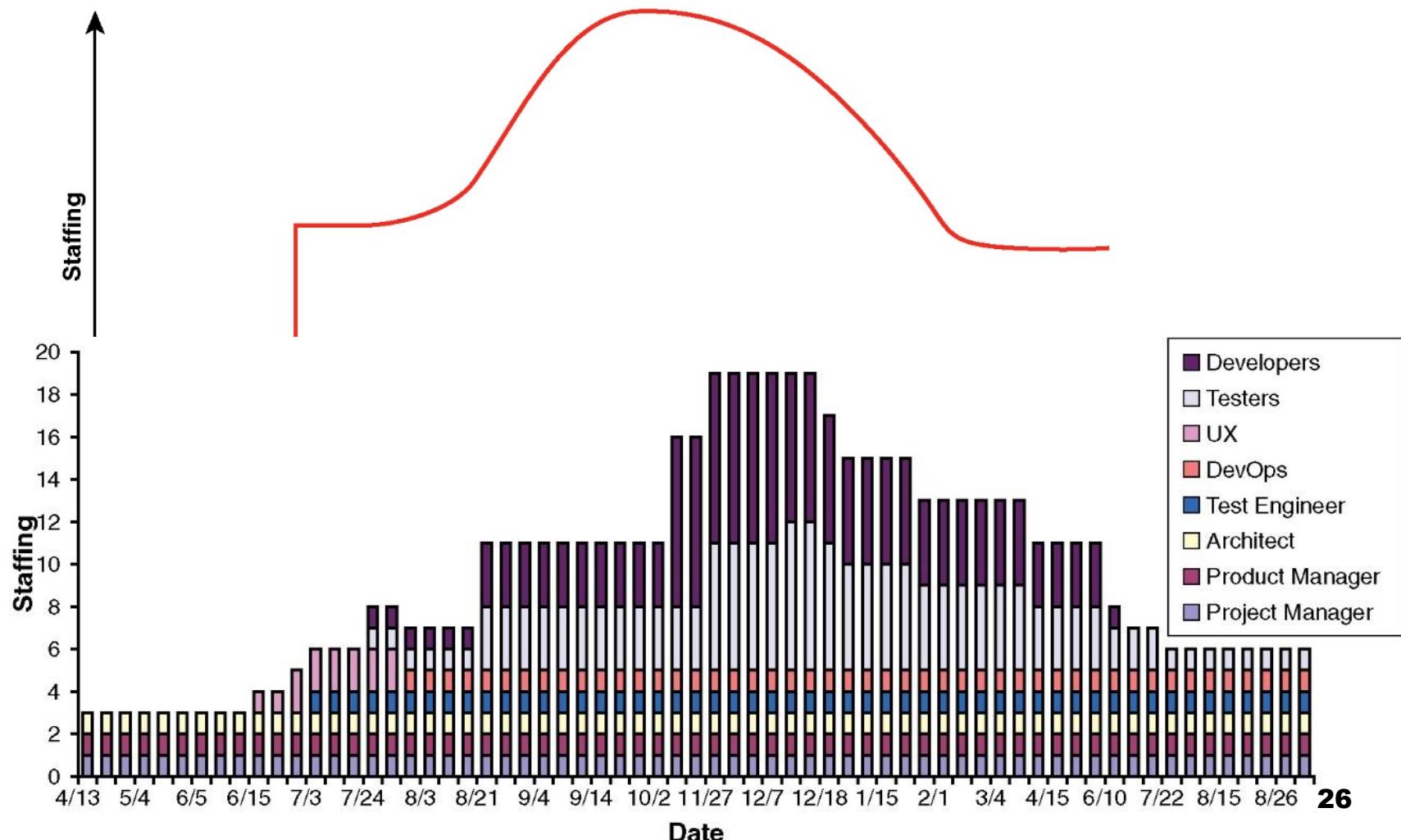
The Core Team = Project Manager, Product Manager, and Architect



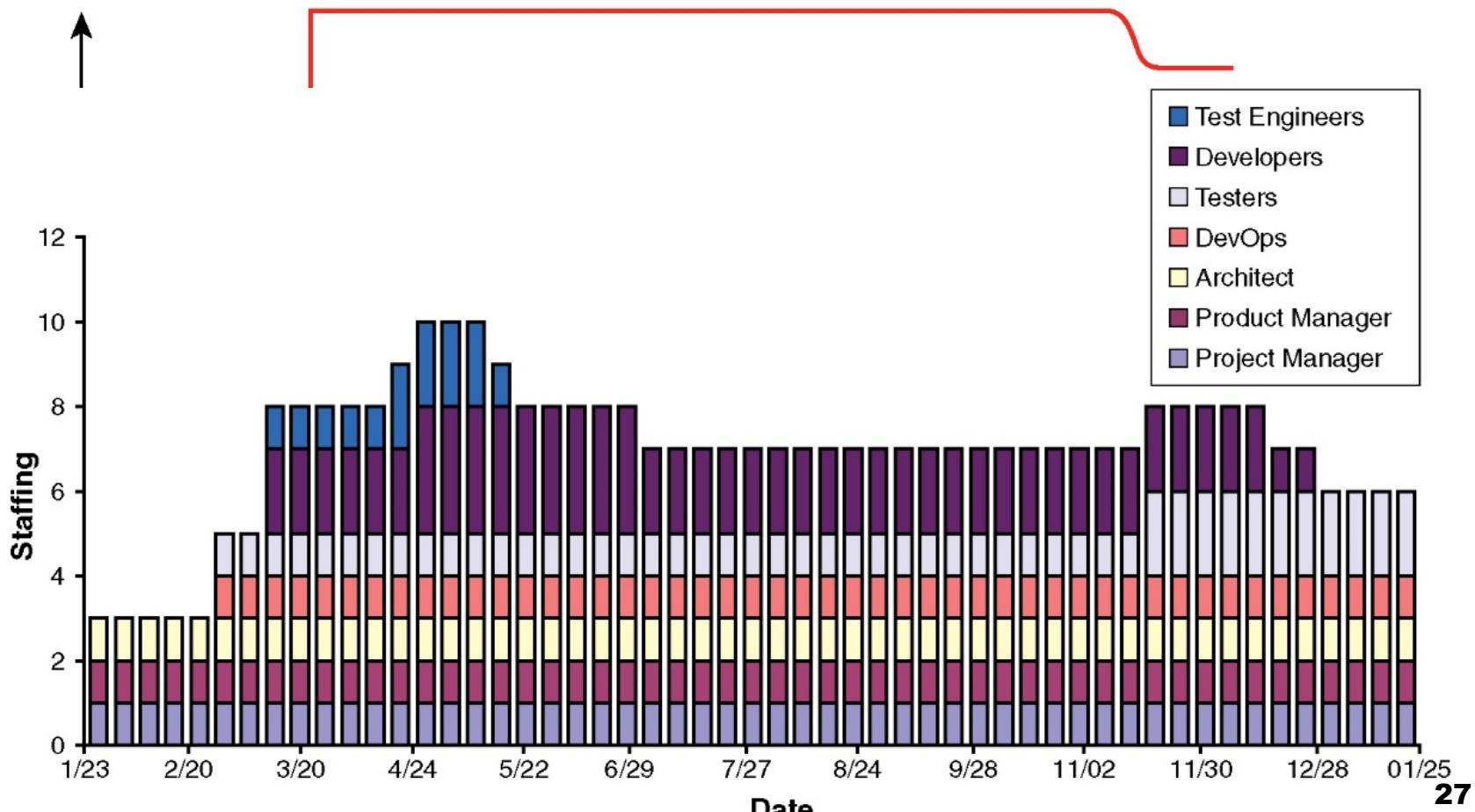
Staffing distribution example



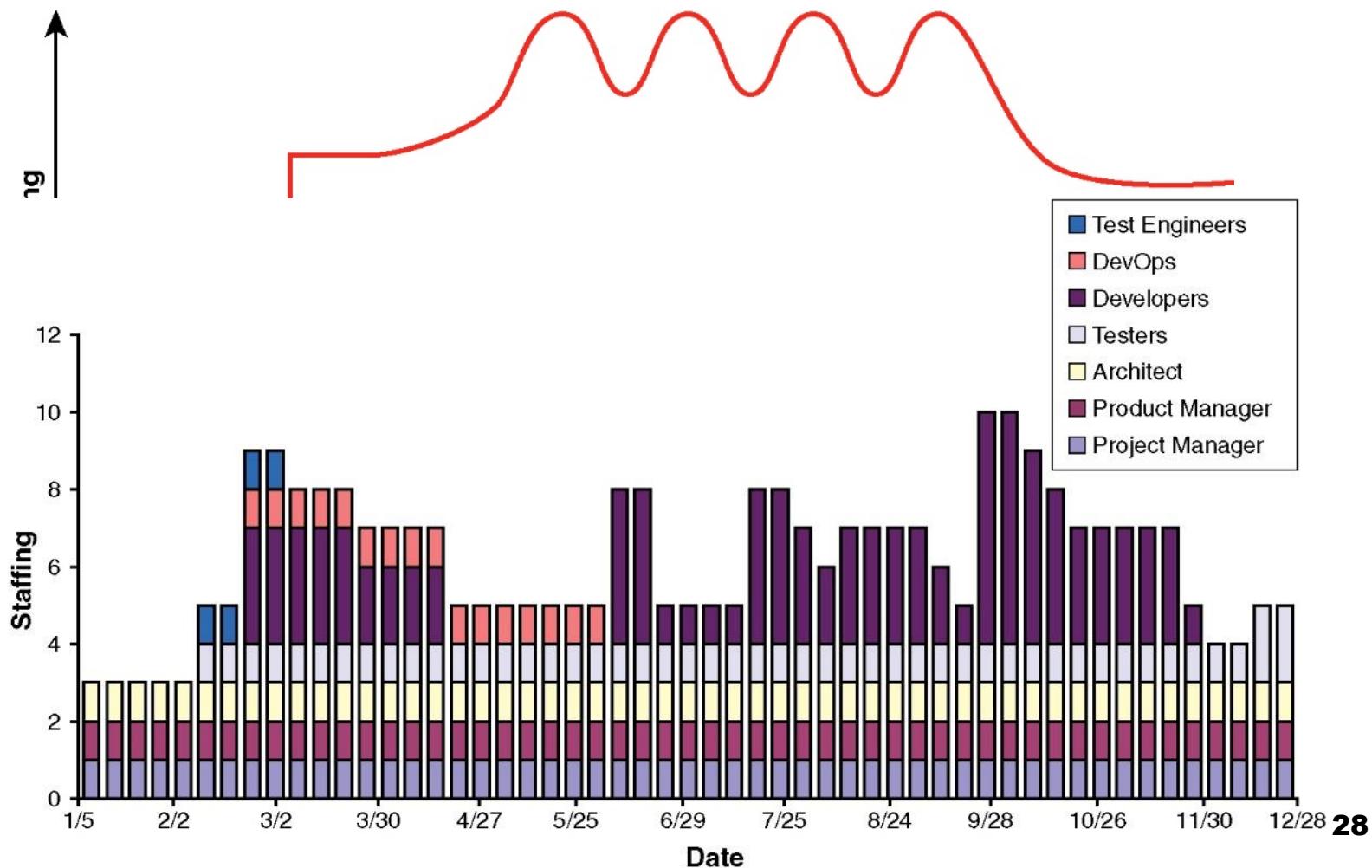
Bad staffing distribution examples - Waste



Bad staffing distribution examples – subcritical staffing



Bad staffing distribution examples - erratic



Bad staffing distribution examples – high ramp-up

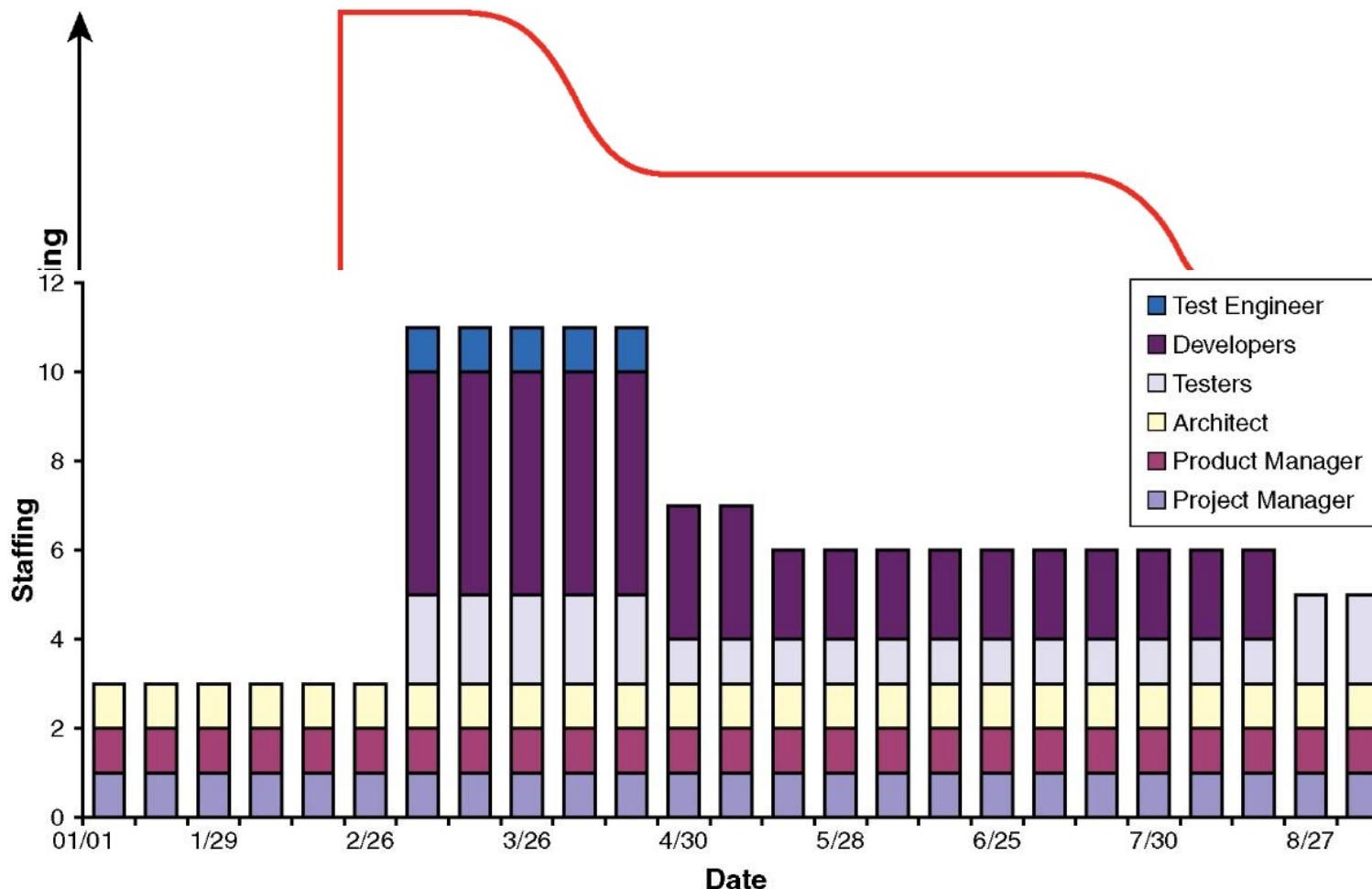
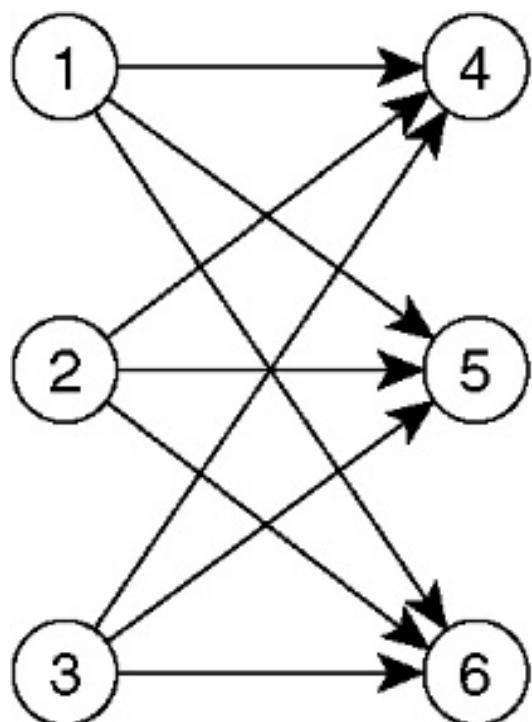
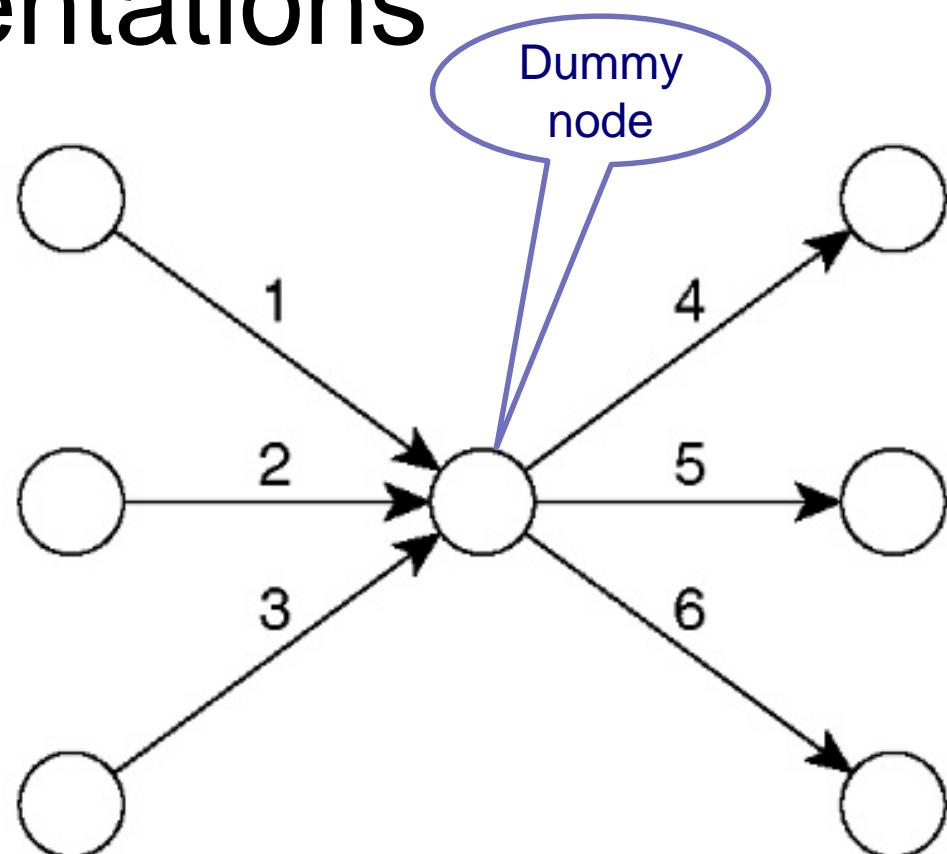


Diagram representations

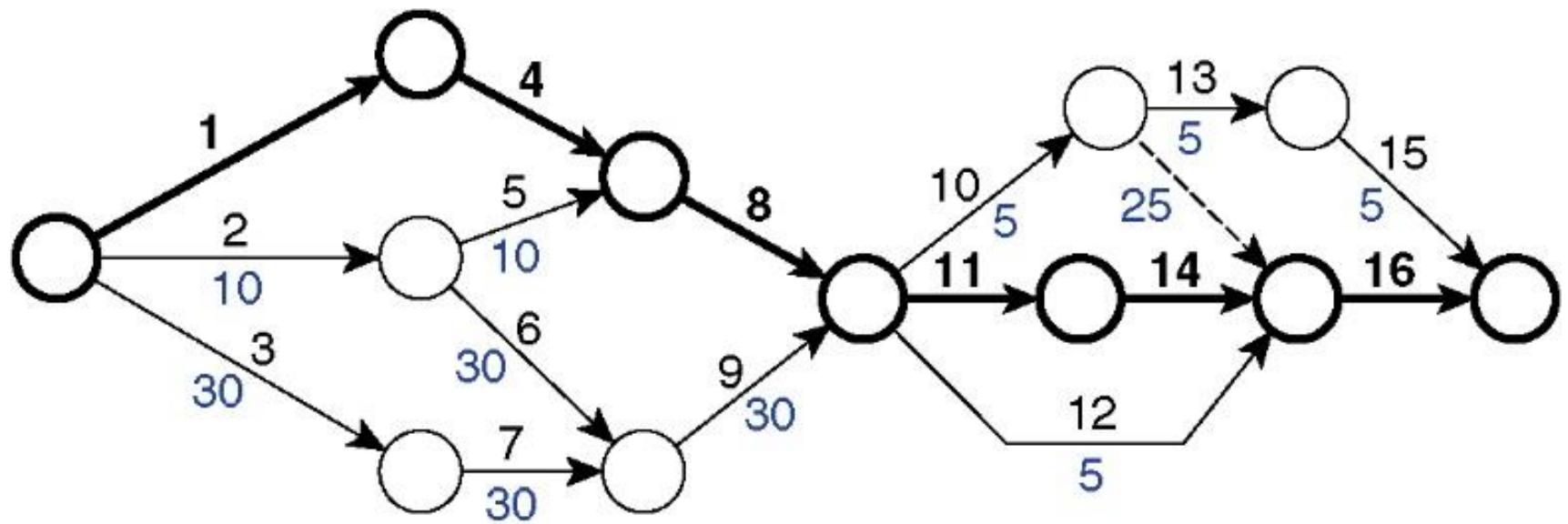


Node Diagram

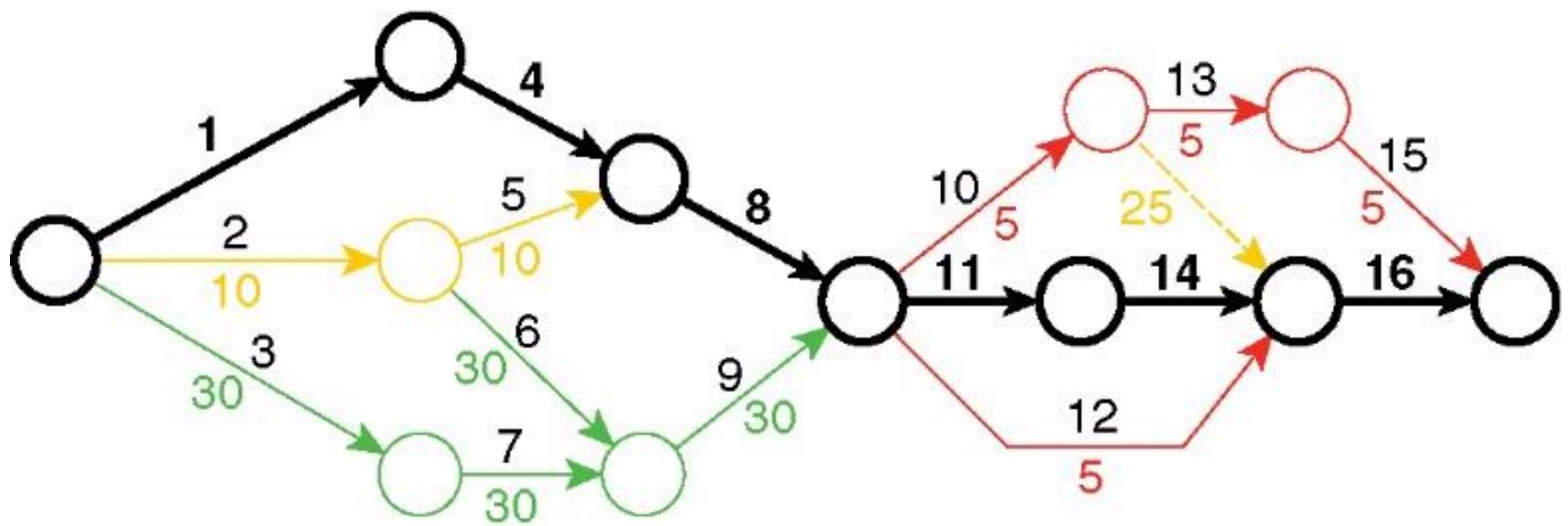


Arrow Diagram

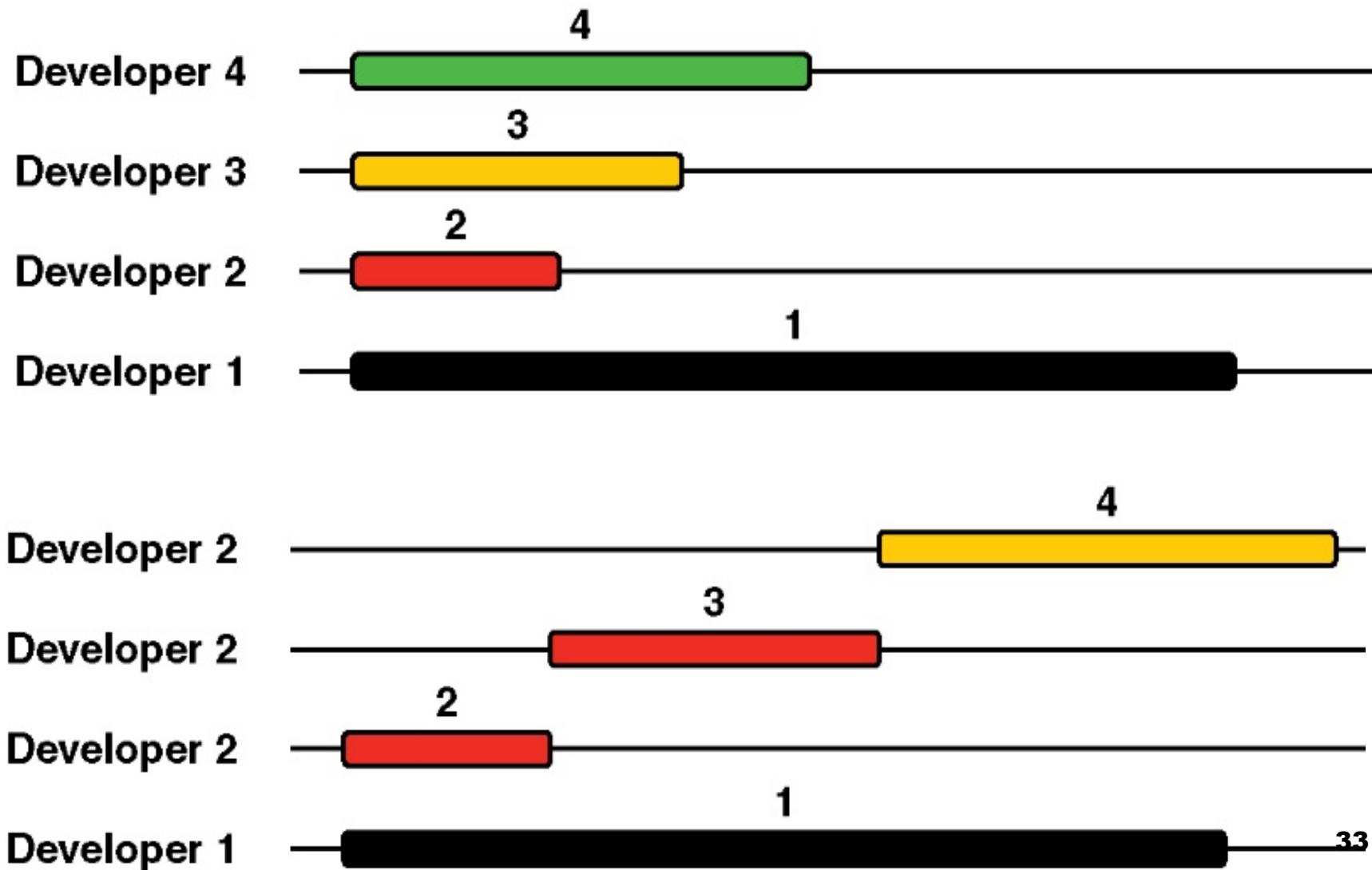
Float color coding



Float color coding



Staffing alternatives



Can we do it faster?

- Project duration bound on CP
- Work smarter, cleaner and correct vs. work harder or add more people
- Best practices
 - Assure **quality**
 - Employ test **engineers**, add testers
 - Invest in **infrastructure**
 - Improve **development skills**
 - Improve the **process**
 - Adopt and employ **standards**

Schedule compression

- Accomplish the same objectives faster
- Two immediate solutions
 - Work with better resources
 - Ex. assign senior developers on critical activities
 - Work in parallel
 - Extract internal phases of an activity and move them elsewhere in the project
 - Remove dependencies between activities

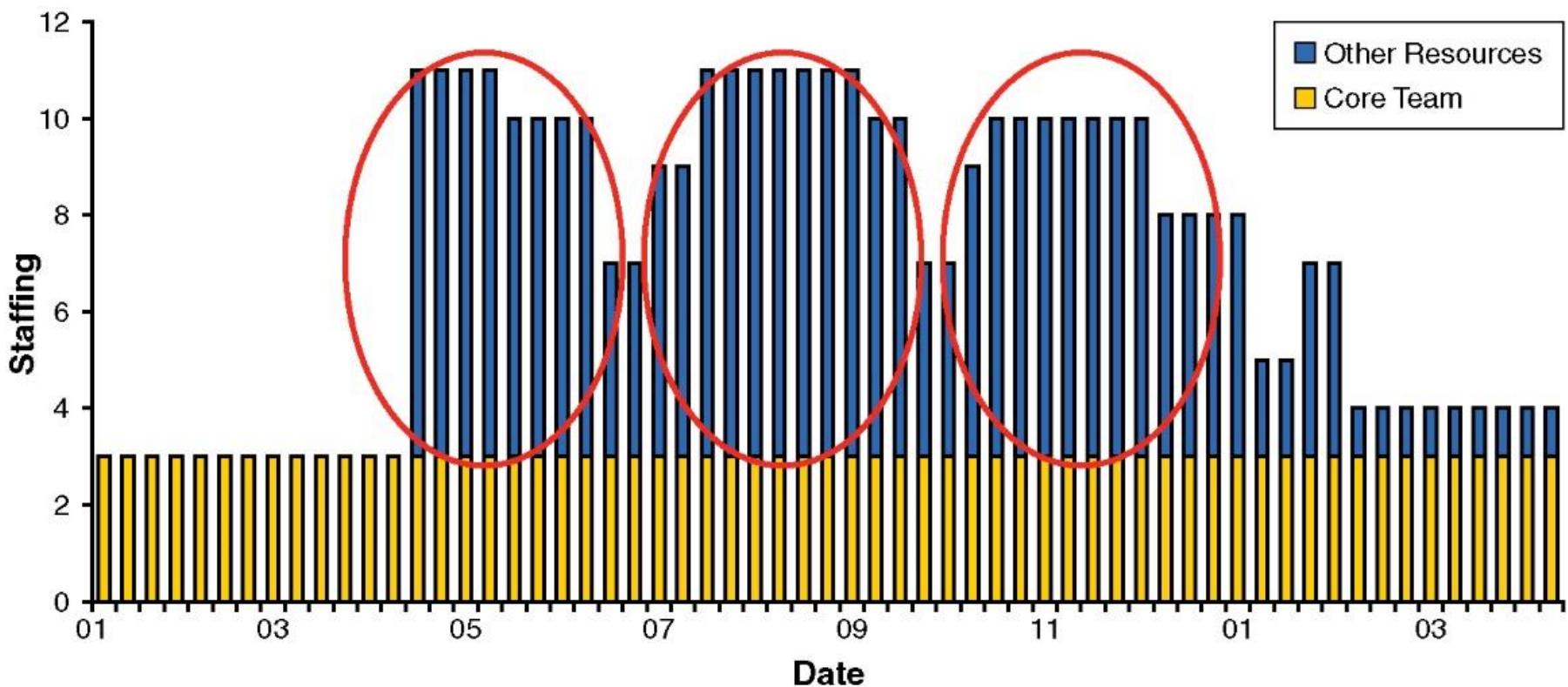
Remove dependencies

- Contract design
- Emulators/simulators development
- Repeated integration and testing
 - ⇒ Additional activities are needed
 - ⇒ Additional resources are needed

Parallel work and cost

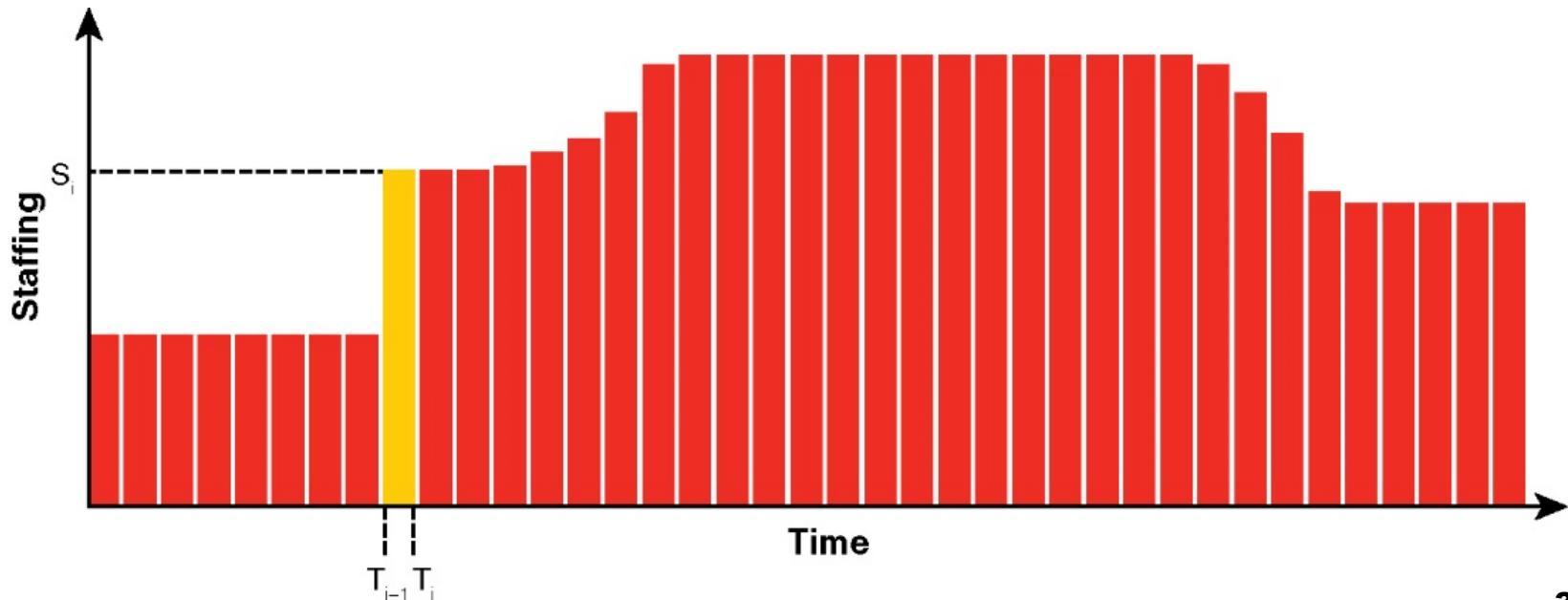
- Parallel work => additional resources
 - ⇒ Increased cost
 - ⇒ Increased noise
 - ⇒ Less efficient execution
 - ⇒ Increased execution complexity
 - ⇒ Increased risk

Candidates for parallel work



Project cost

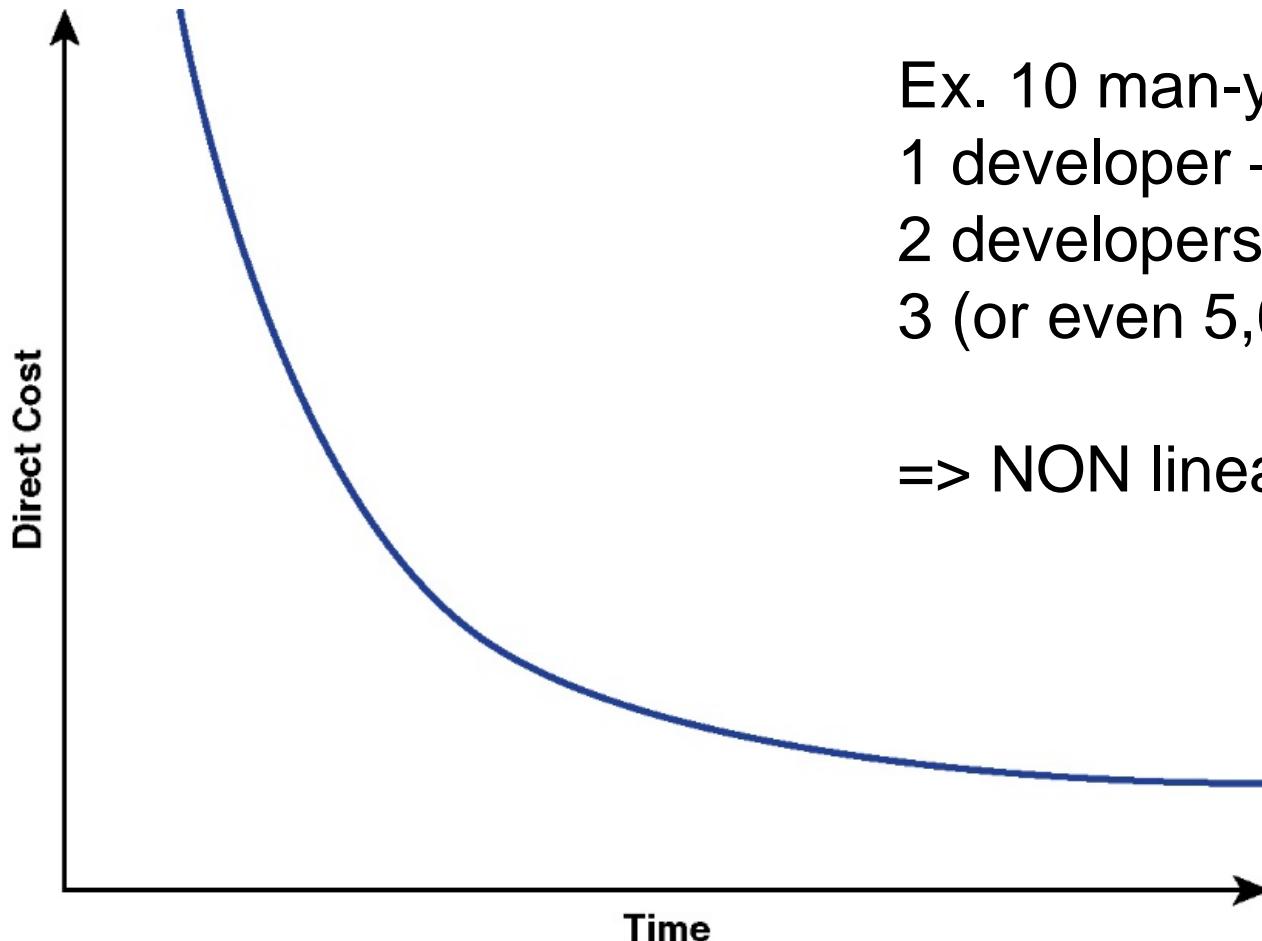
- Cost = Staffing * time
- Cost = $\sum(S_i * (T_i - T_{i-1}))$



Project efficiency

- **efficiency** = sum of effort across all activities (assuming perfect utilization of people)/actual project cost.
- Ex. estimated effort = 10 man-months,
 actual cost = 50 man-months =>
 Efficiency = 20%
- Real well-designed projects have efficiency between 15%-25%

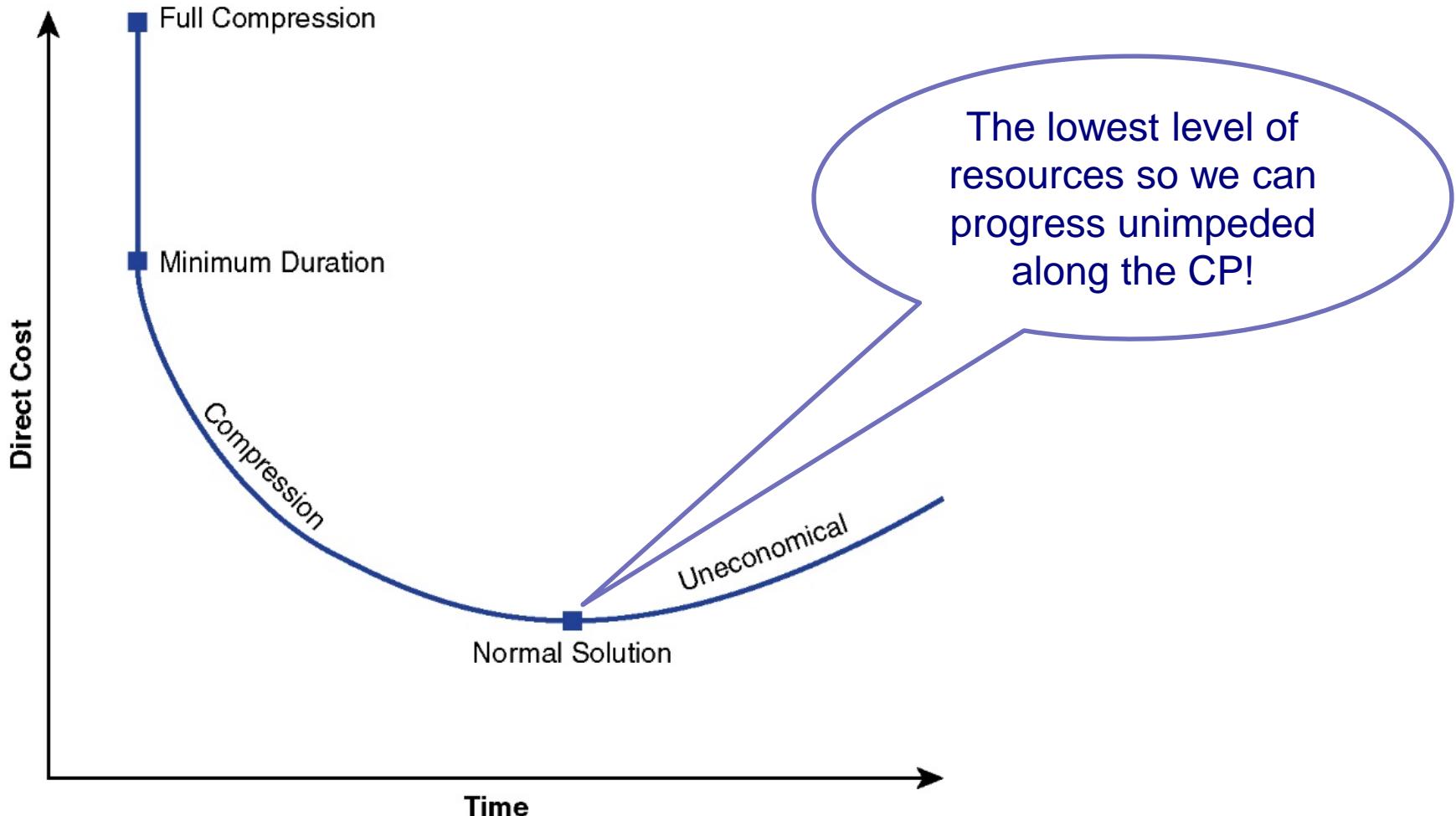
Idealized time–cost curve



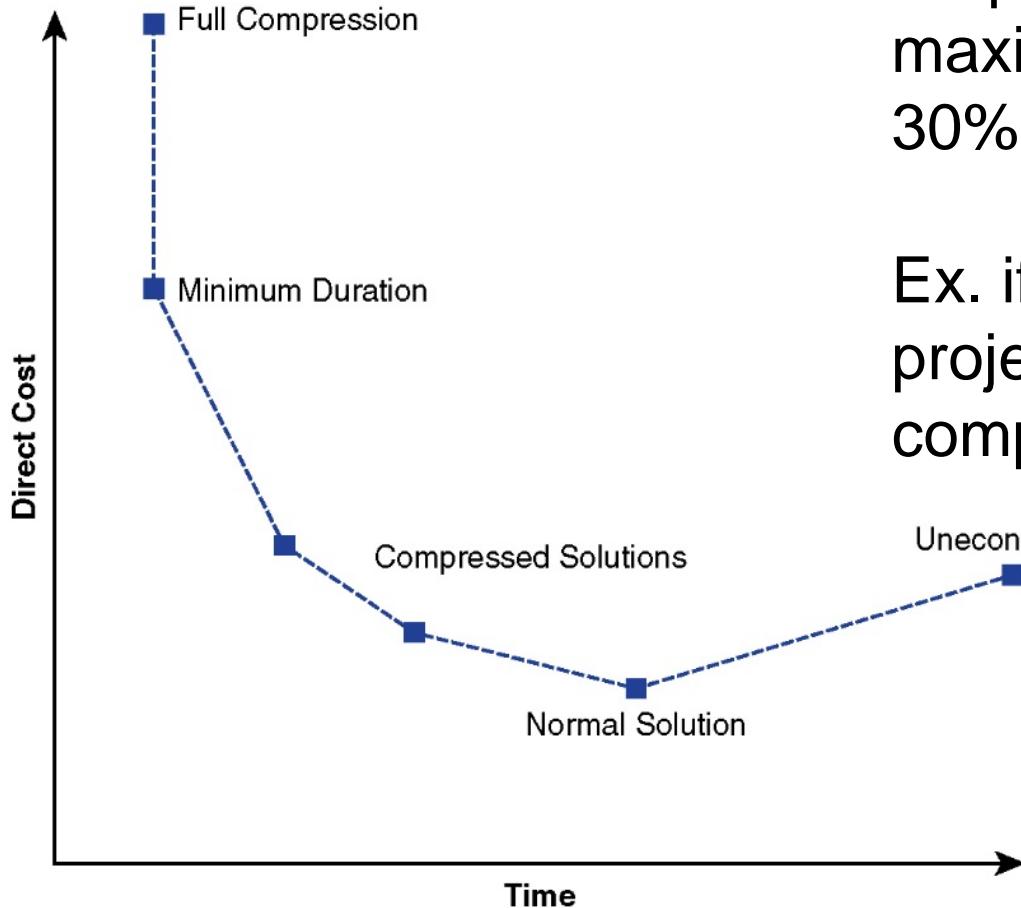
Ex. 10 man-years
1 developer – 10 years
2 developers – 7 years (not 5!)
3 (or even 5,6 devs) – 5 years

=> NON linear!

Actual time–cost curve



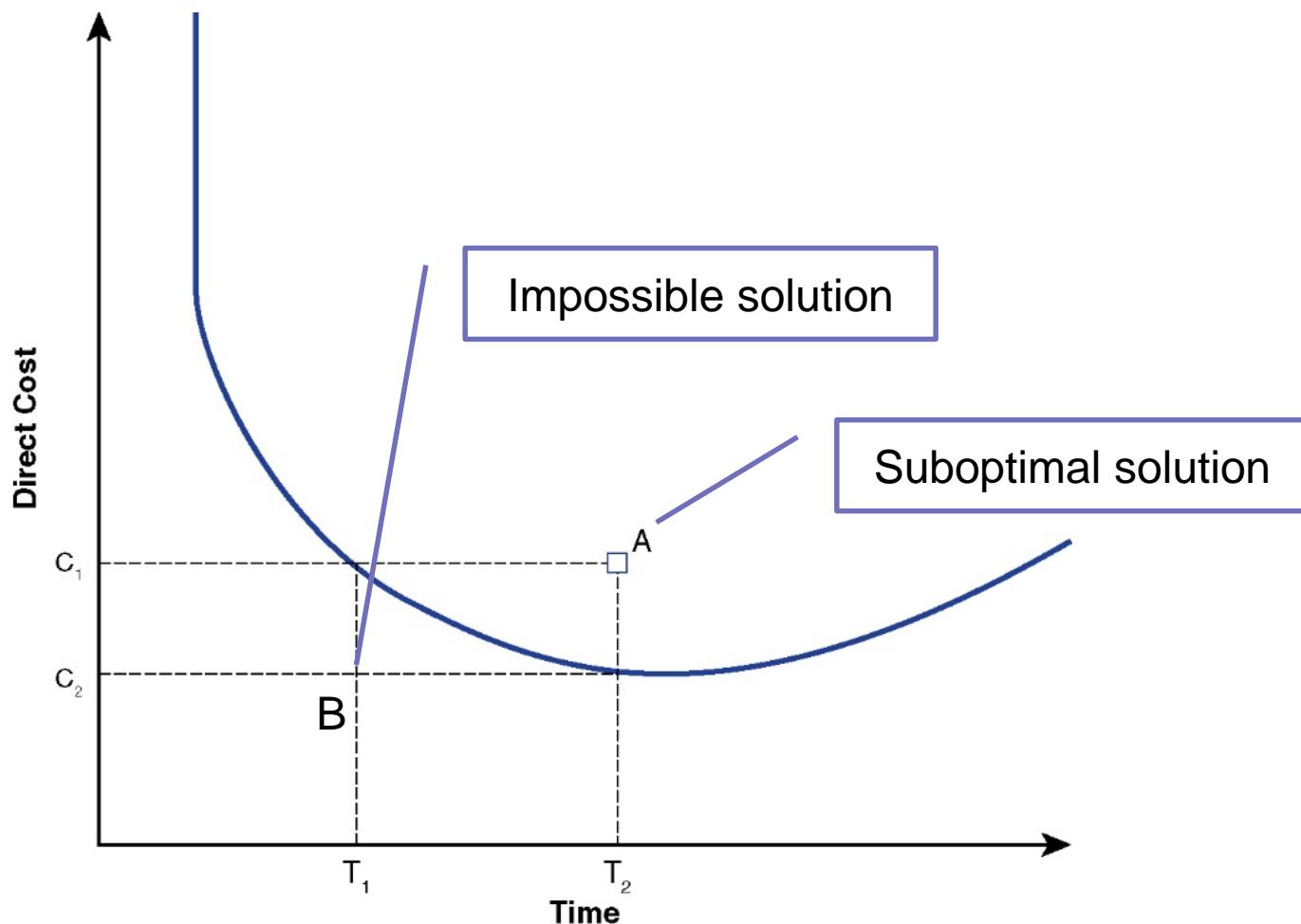
Discrete time–cost curve



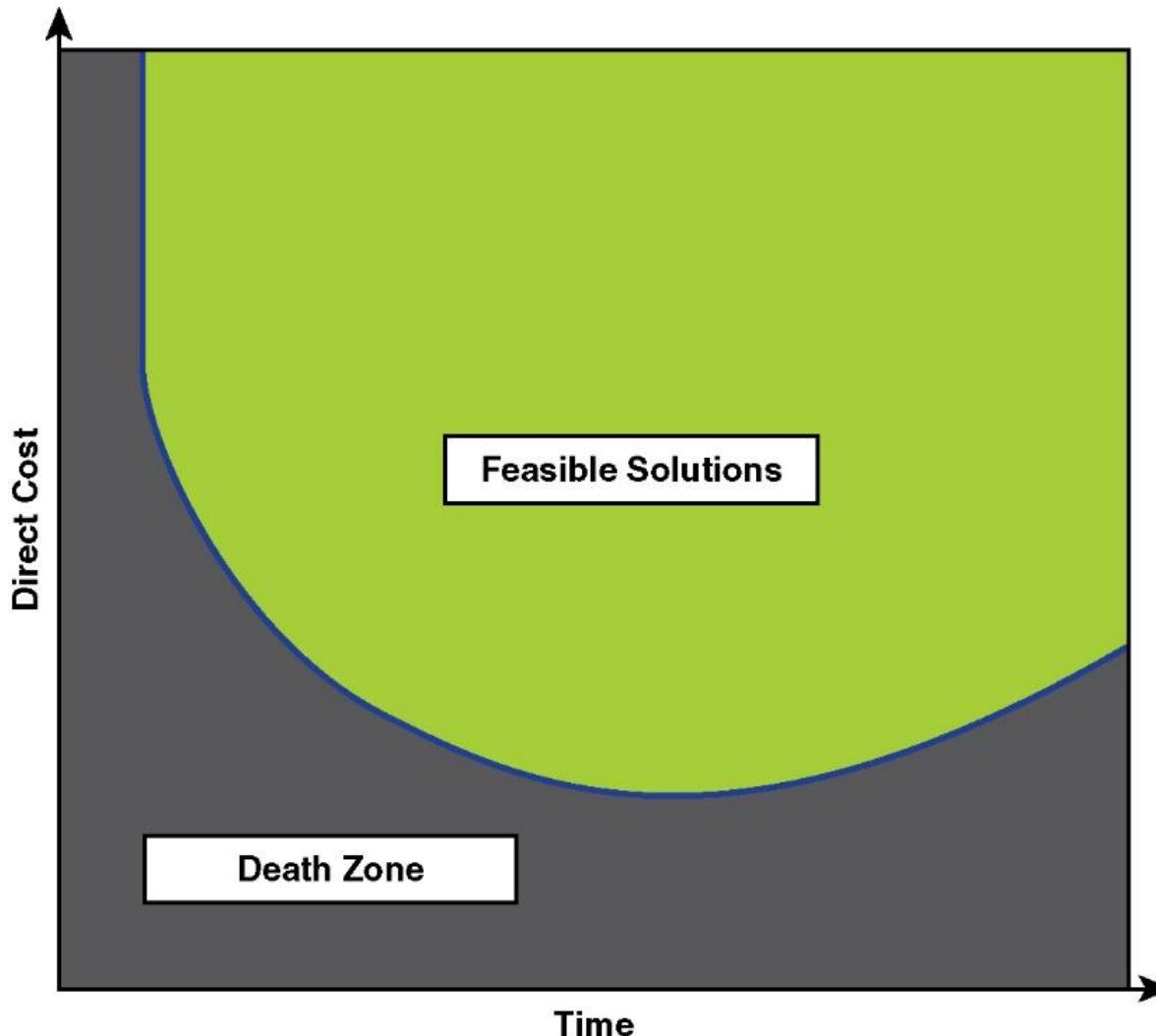
Empirical finding: the maximum compression limit = 30% of the normal solution

Ex. if the normal solution of a project = 12 months it can be compressed at most to 8.4

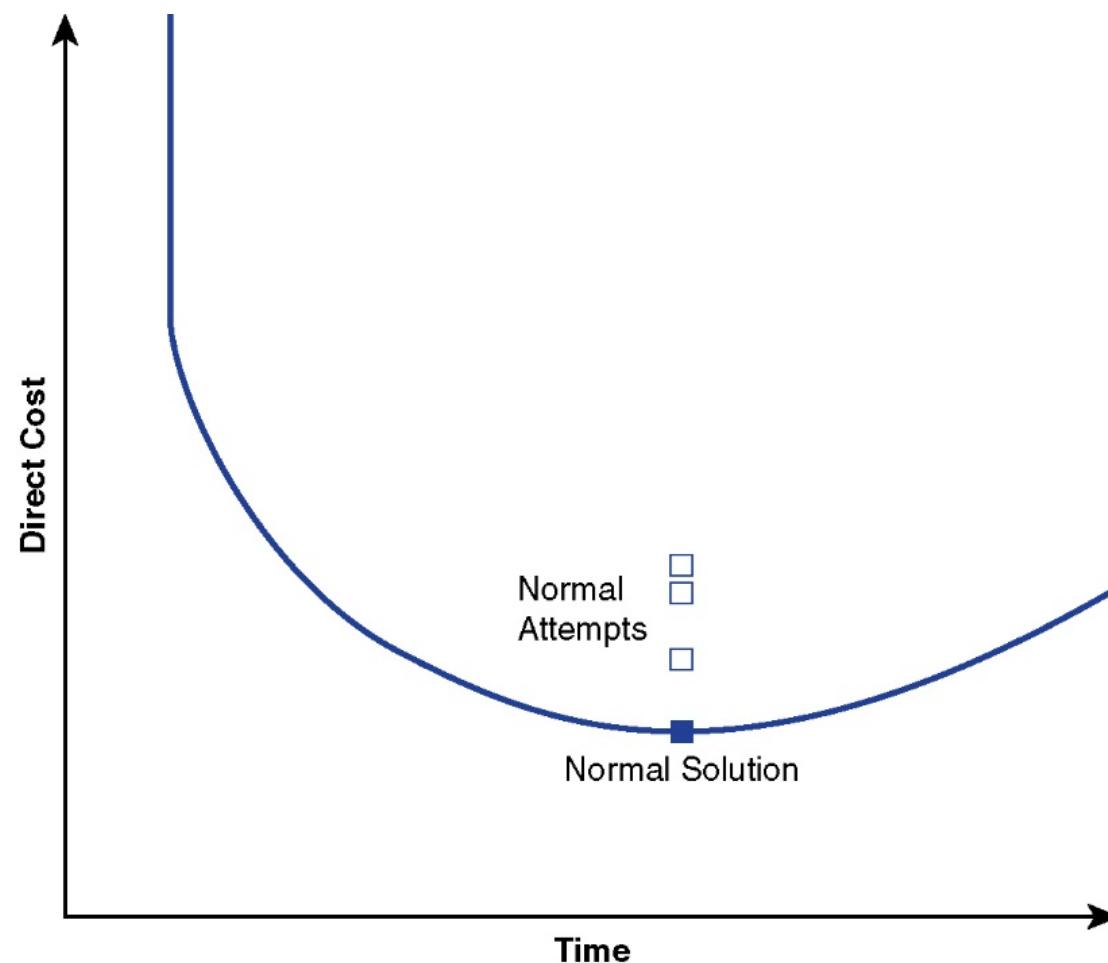
Project feasibility



Avoid the death zone



Example on finding the normal solution



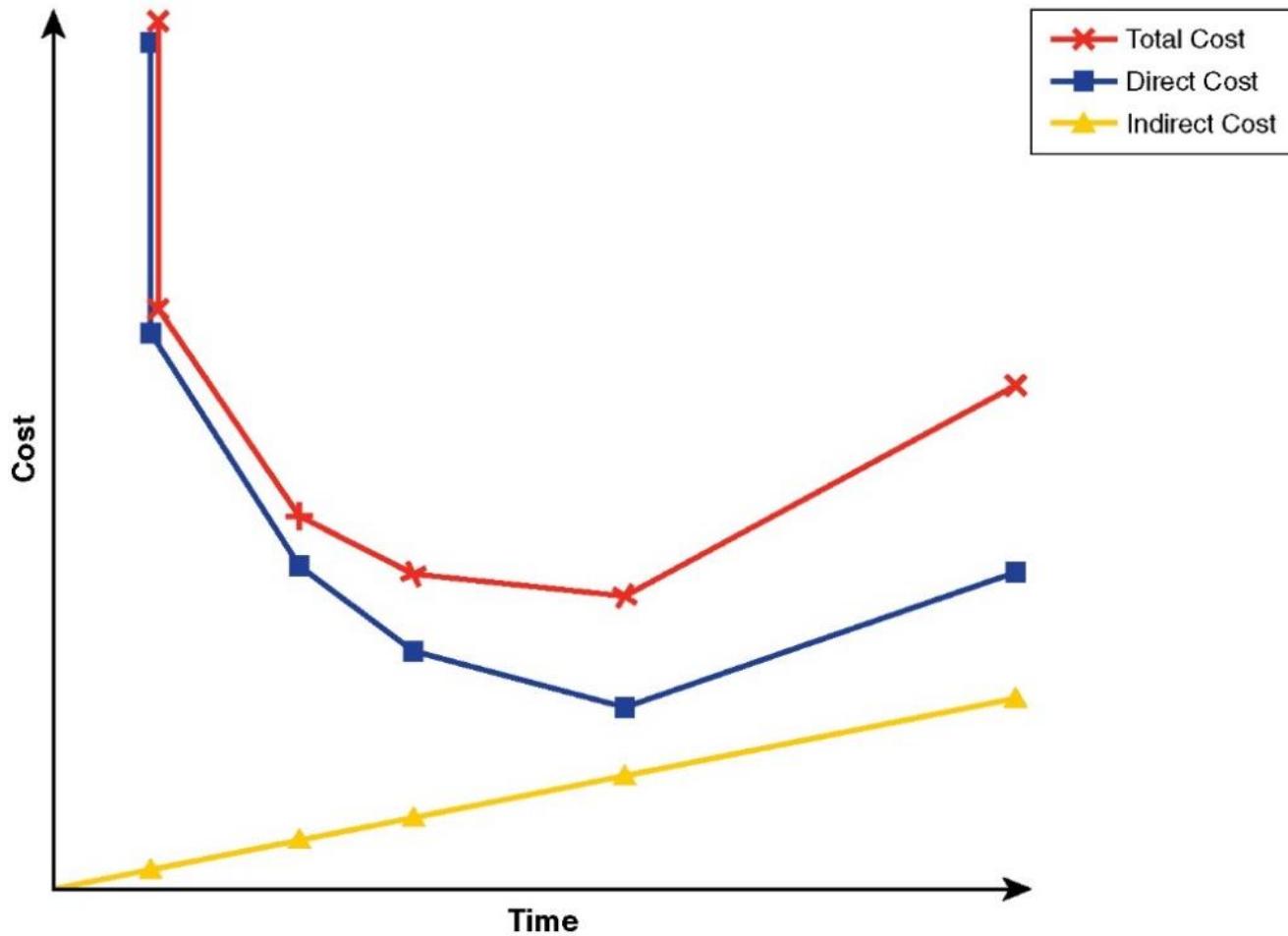
Project cost elements

- Direct cost - activities that add direct measurable value to the project
 - **Developers** working on services
 - **Testers** performing system testing
 - **Database architect** designing a database
 - **Test engineers** designing and building a test harness
 - **UI/UX experts** designing the user interface
 - **The architect** designing the system or the project

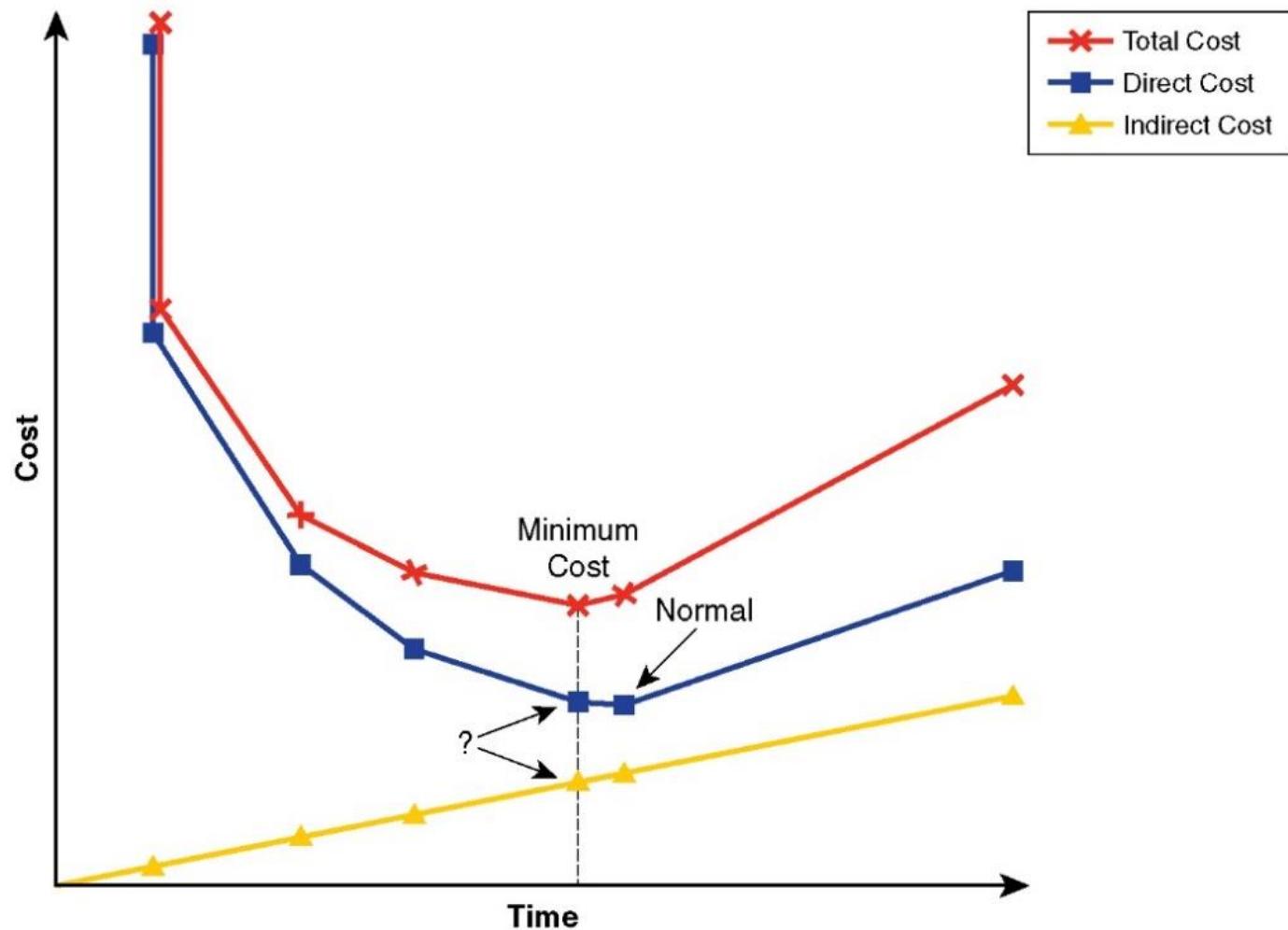
Project cost elements

- Indirect cost - activities that add indirect immeasurable value to the project
 - The core team (i.e., architect, project manager, product manager) after the SDP review
 - Ongoing configuration management, daily build and daily test, or DevOps in general
 - Vacations and holidays
 - Committed resources between assignments

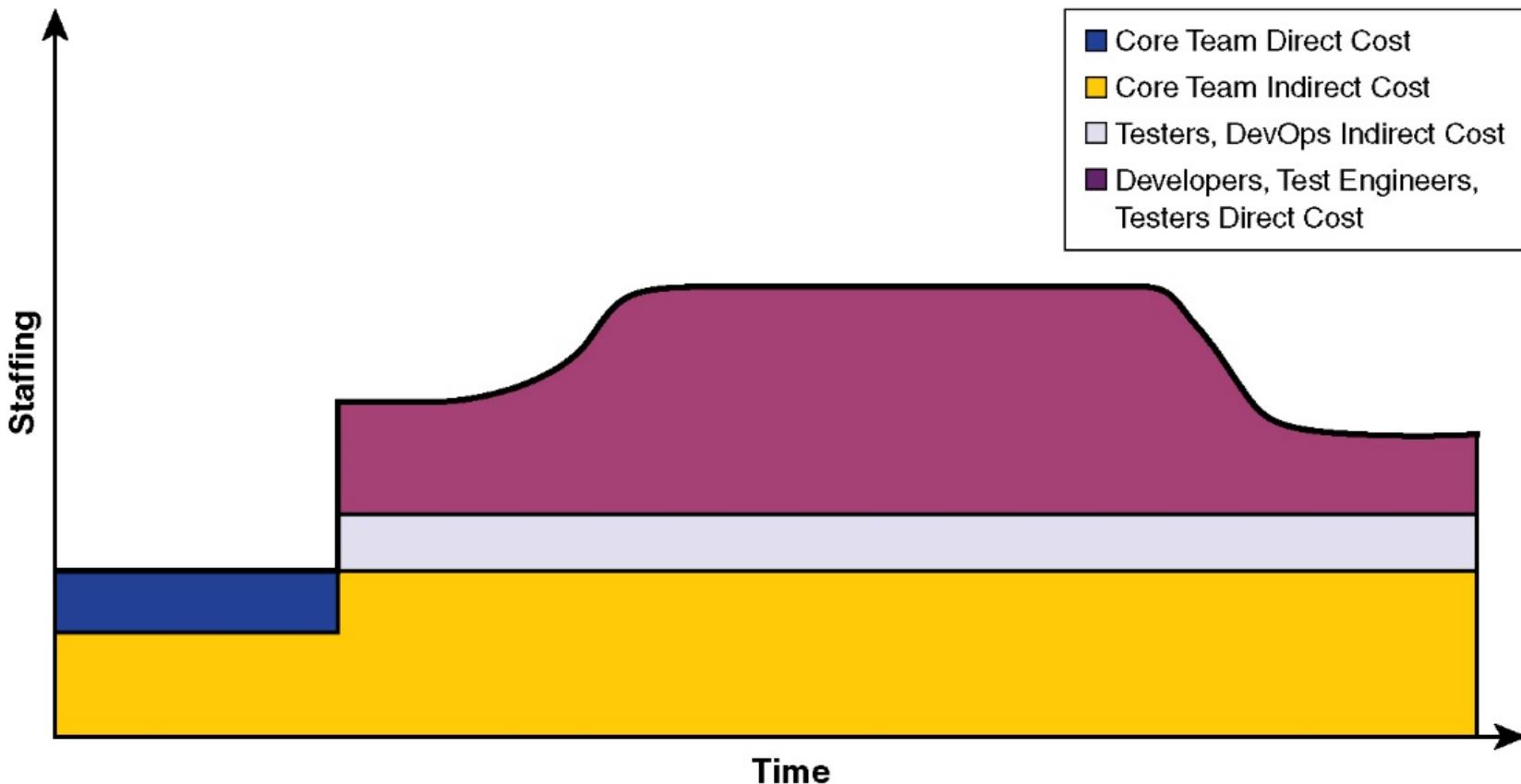
Cost elements

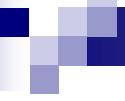


Shifts of minimum cost



Cost elements under the staffing distribution

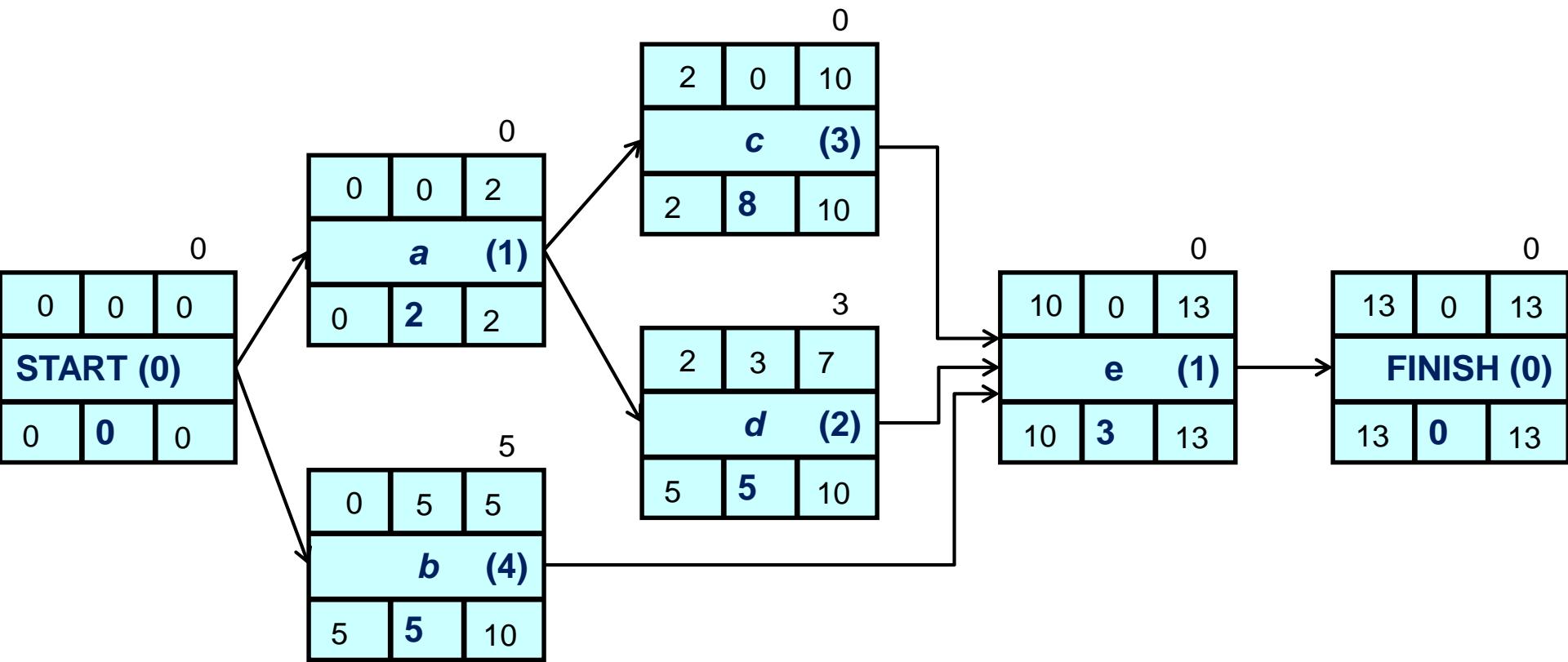




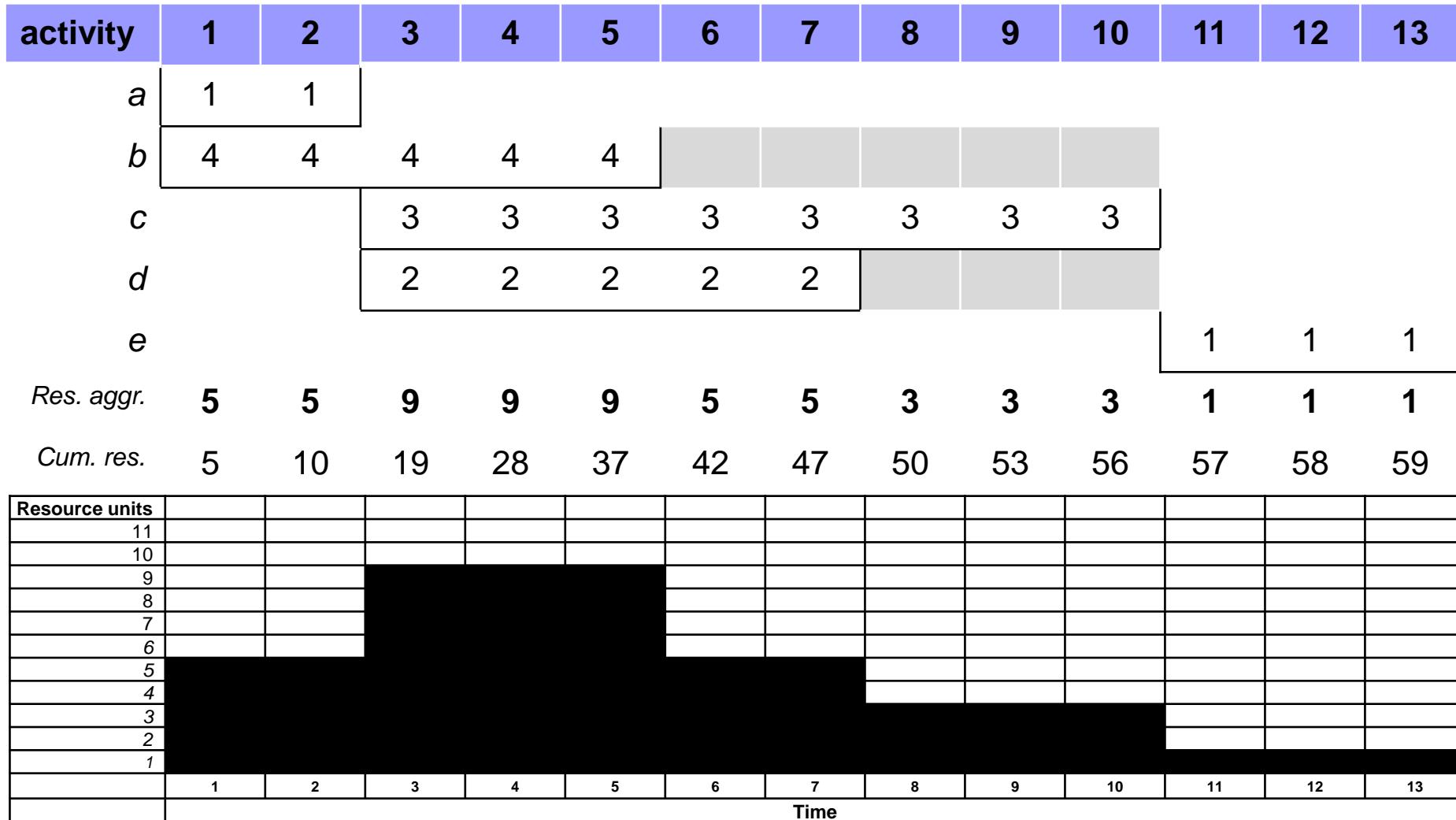
Generalization Exercise

- What if we need several resources per task?

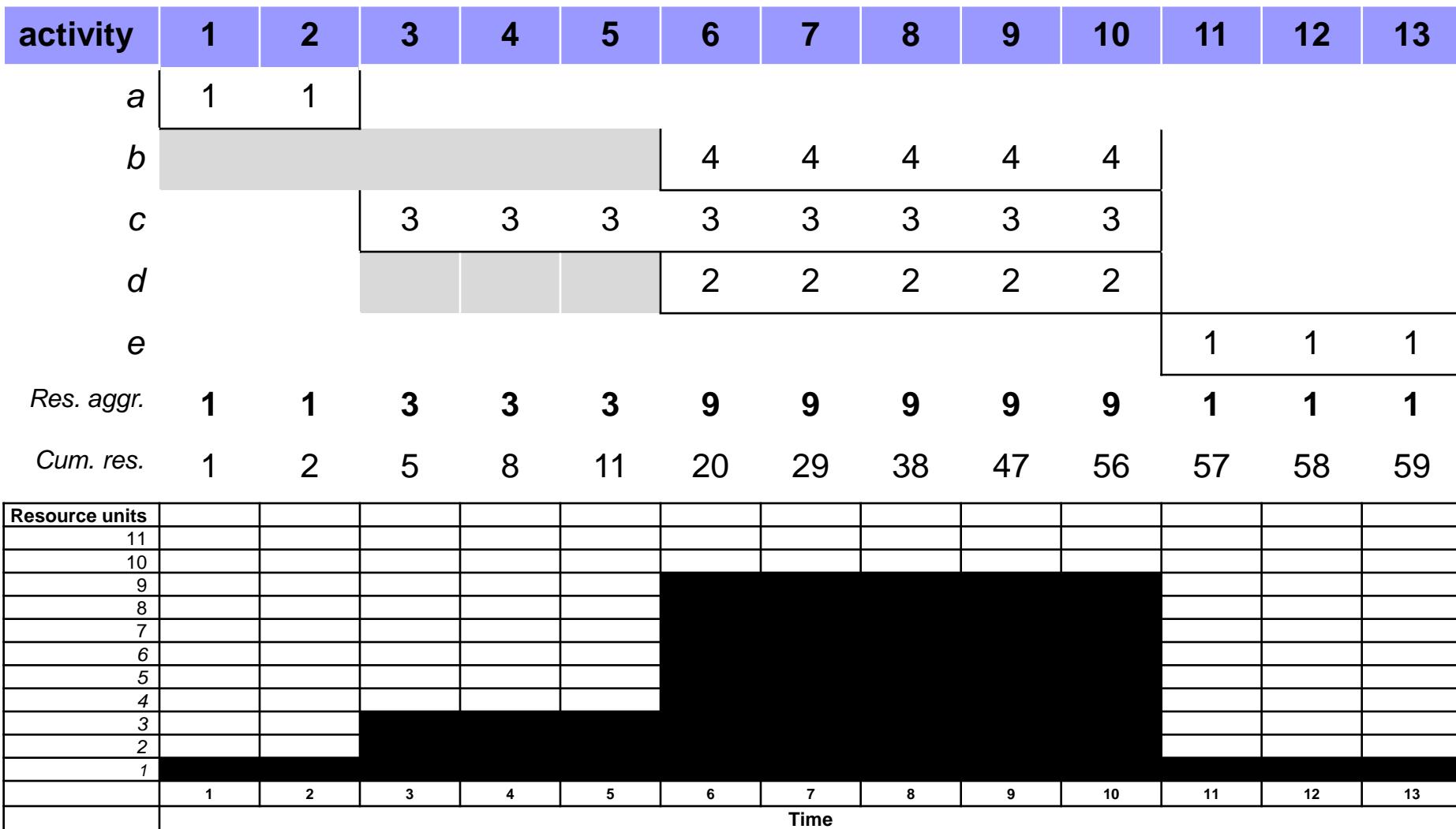
Network example



Aggregation with a bar chart (earliest start)

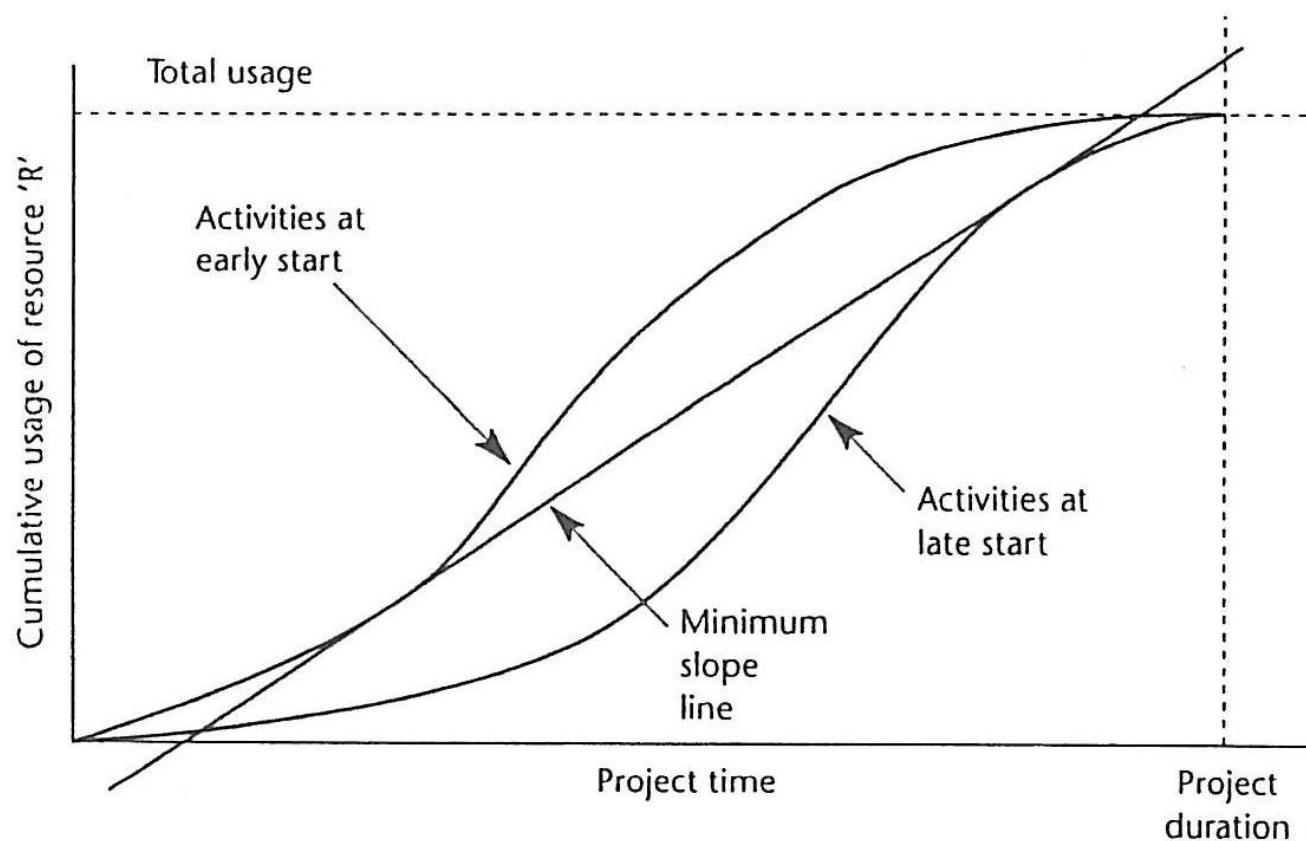


Aggregation with a bar chart (latest start)

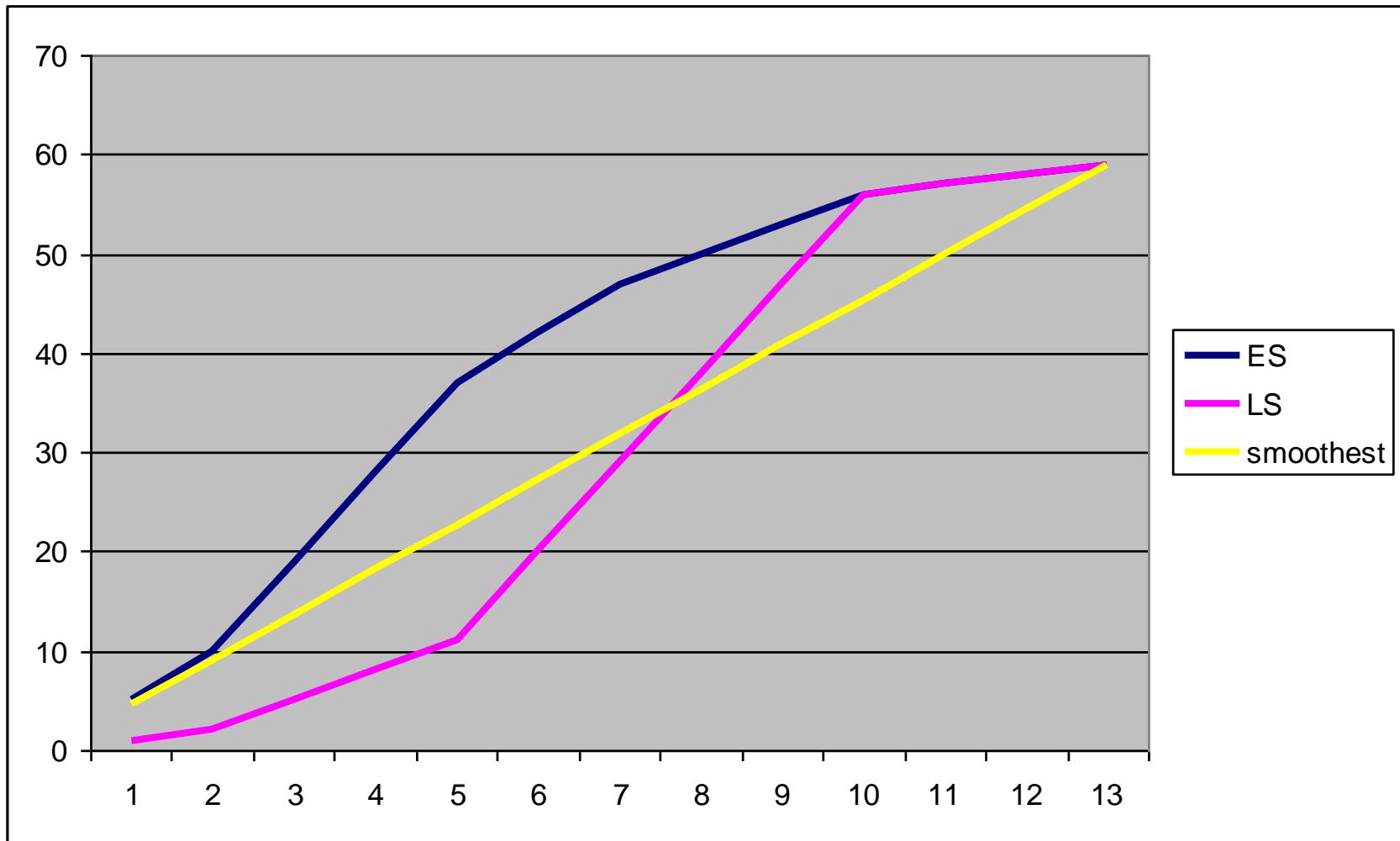


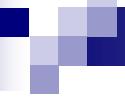
The S Curve analysis

- The minimum slope level is the less ‘critical’ from the viewpoint of availability



S Curve of the example

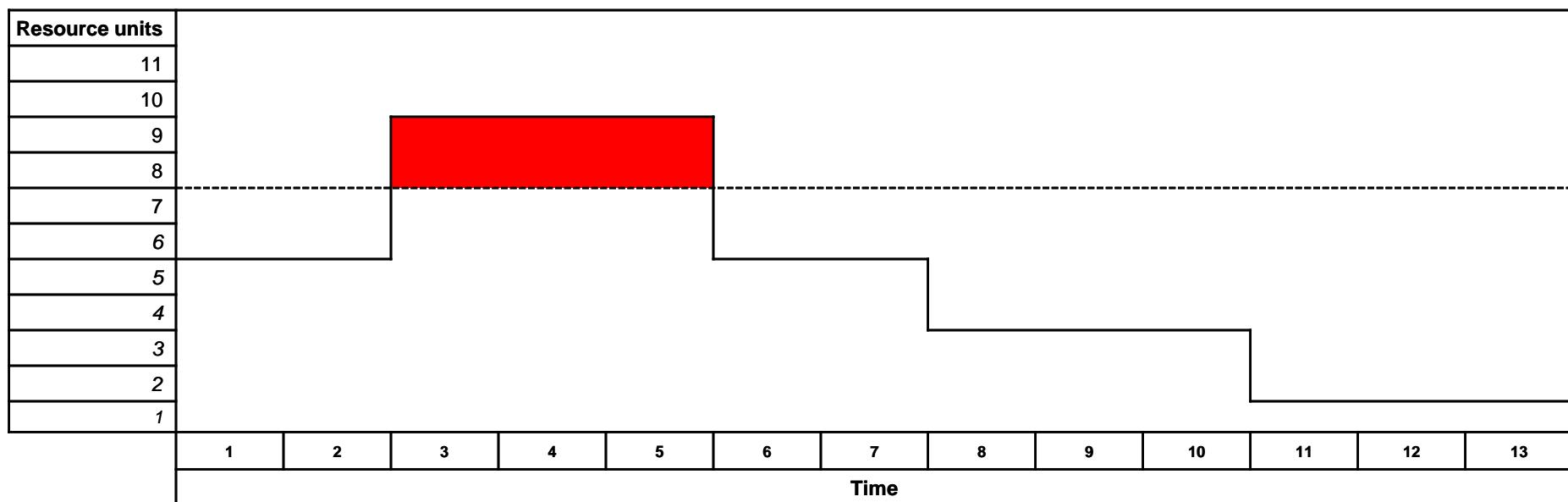




Other possibilities

- Alternative resources
- Alternative methods
- Alternative sequences
(if there is no technical dependency)

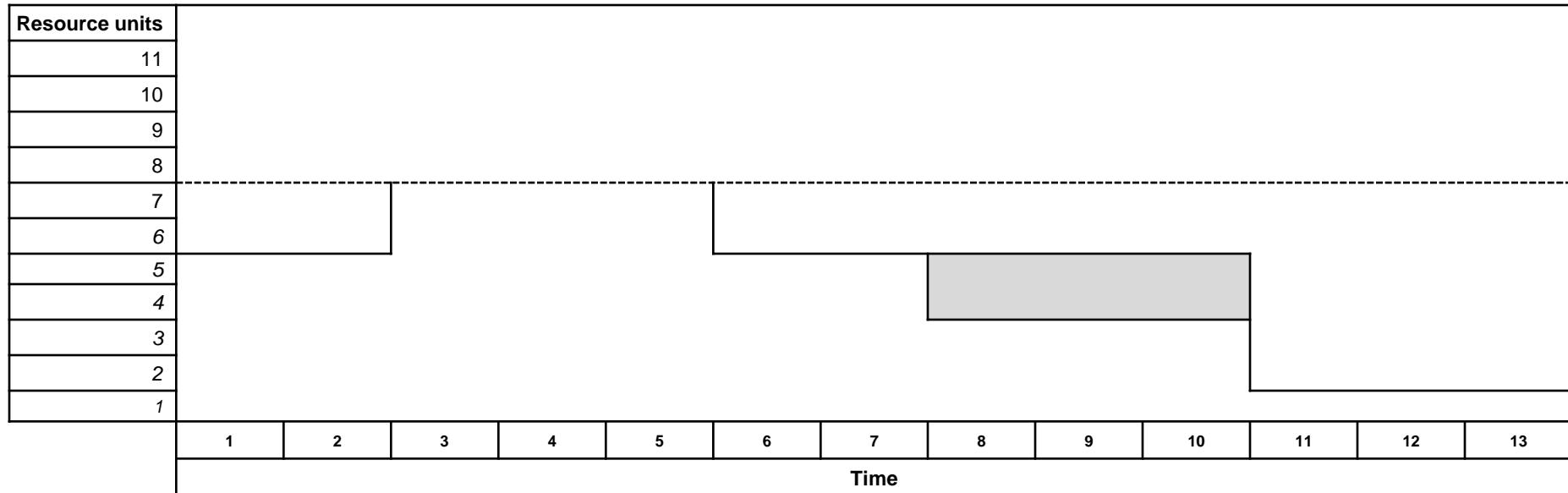
Levelling the load



- Trying to keep the original duration unchanged means that critical activities should not be moved. Thus try to move activities with free float.

Solution

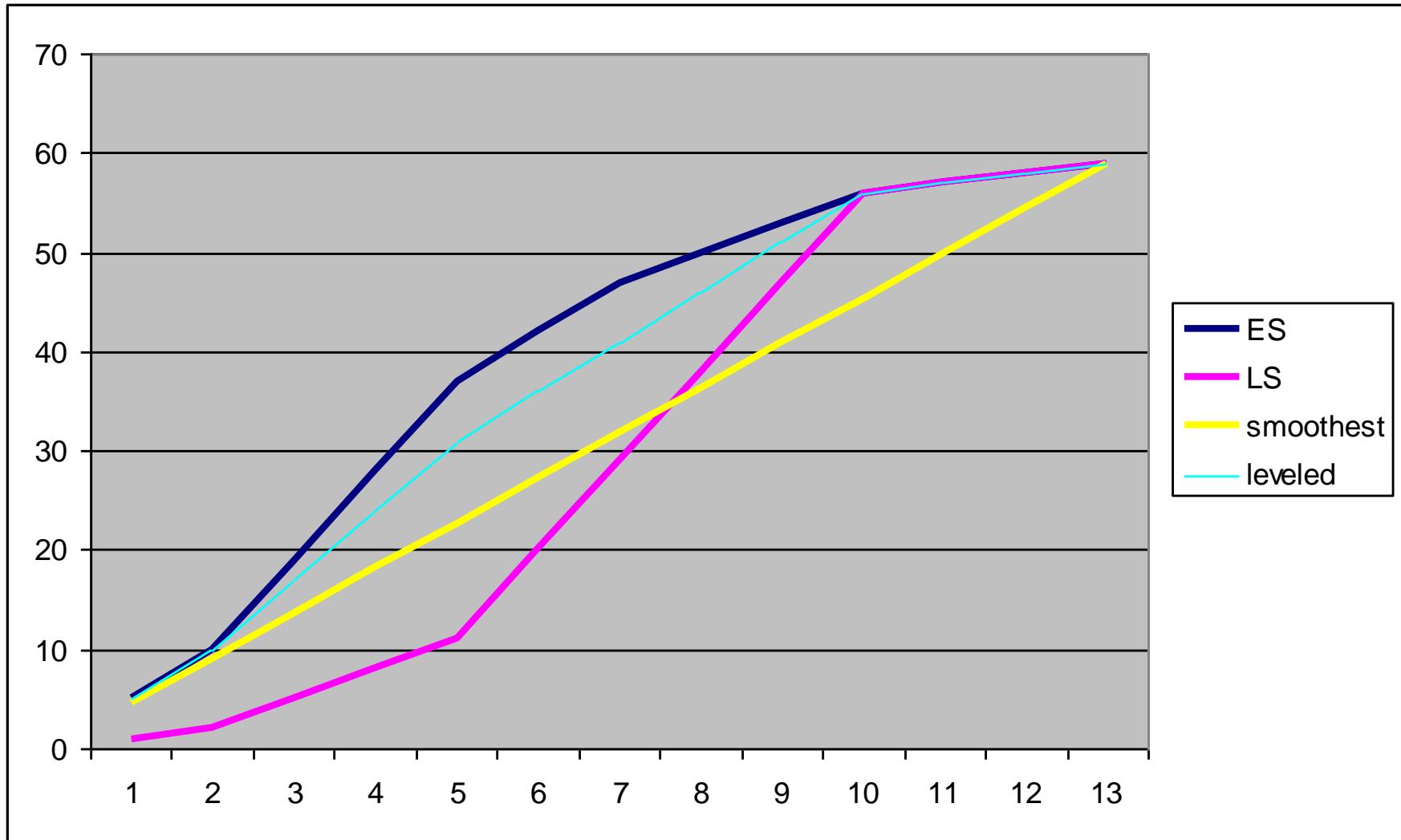
- There are only 2 activities with free float: b , d
 - *Which one to move and to where?*
 - Moving activity d 3 days later is eliminating the peak.



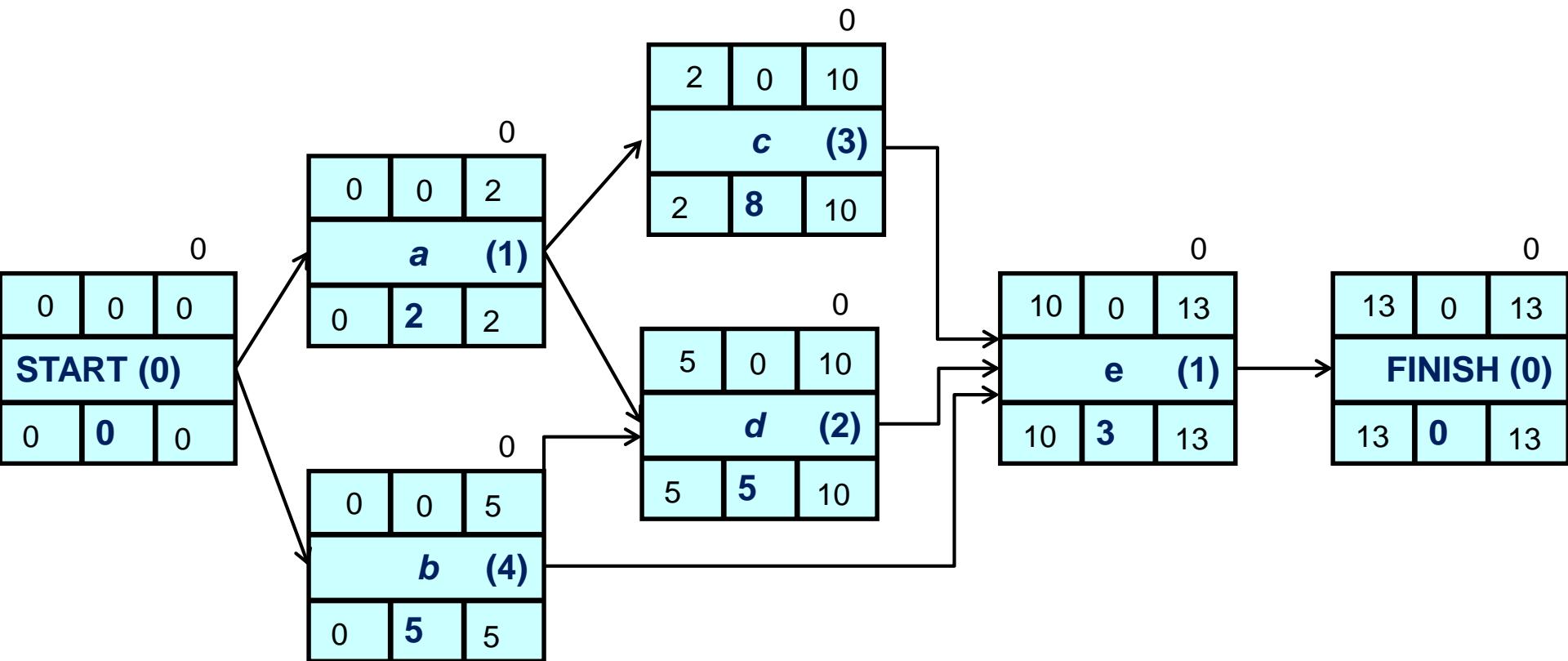
activity	1	2	3	4	5	6	7	8	9	10	11	12	13
<i>a</i>	1	1											
<i>b</i>	4	4	4	4	4								
<i>c</i>		3	3	3	3	3	3	3	3	3			
<i>d</i>		2	2	2	2	2							
<i>e</i>											1	1	1
Res. aggr.	5	5	9	9	9	5	5	3	3	3	1	1	1
Cum. res.	5	10	19	28	37	42	47	50	53	56	57	58	59

activity	1	2	3	4	5	6	7	8	9	10	11	12	13
<i>a</i>	1	1											
<i>b</i>	4	4	4	4	4								
<i>c</i>		3	3	3	3	3	3	3	3	3			
<i>d</i>	-	-	-	-	2	2	2	2	2	2			
<i>e</i>											1	1	1
Res. aggr.	5	5	7	7	7	5	5	5	5	5	1	1	1
Cum. res.	5	10	17	24	31	36	41	46	51	56	57	58	59

S Curve with leveled solution



Effect of levelling



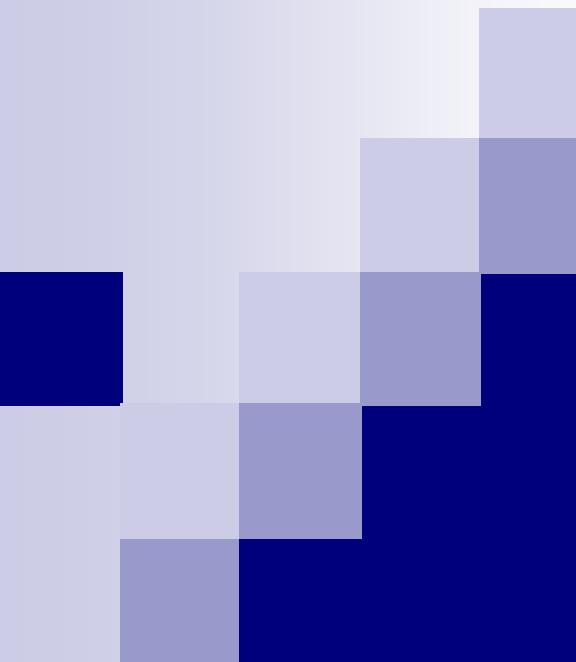
- Changes: new precedence relationship, more critical paths => increased risk!

Wrap-up

- Resource management
 - Iterative resource allocation
 - Resource distribution
 - Resource costs
- Schedule compressing and cost trade-off

Next time discussion topic

- Consider a network diagram including tasks durations, dependencies and resources per activity. Calculate the CP and find the normal solution (i.e. the needed human resources and for how long). Show your reasoning and the final resource distribution diagram.
- Extra challenge – try to shorten the normal solution. Show the updated solution



Risk Management

Lecture 8

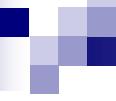
Risk Management Topics

■ Risk Management

- Principles
- Process
- Indicators

■ References

- *MCSD Self-Paced Training Kit: Analysing Requirements and Defining Microsoft .NET Solution Architectures*, Microsoft Press
- *Managing Iterative Software Development Projects*, Kurt Bittner, Ian Spence, Addison Wesley 2006
- RUP
- Righting Software by Juval Löwy, 2020 Pearson Education (Chapter 10)

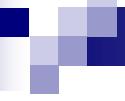


Primary SDP themes

- Getting the design right by focusing on architecture first
- **Managing risk through iterative development**
- Reducing complexity by using components and services
- Making software progress and quality tangible through instrumented change management
- Automating the overhead and bookkeeping by using automation and round-trip engineering tools

Risk Management

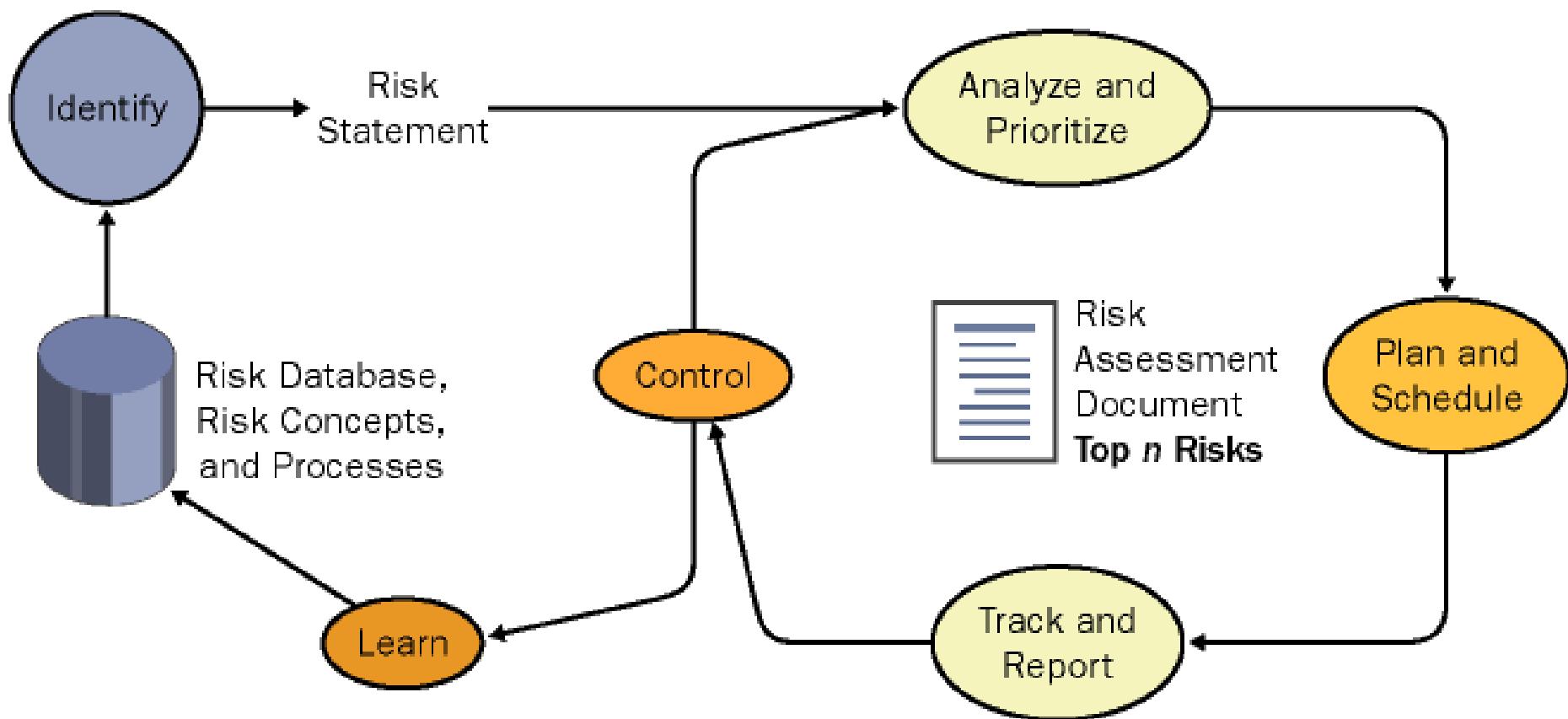
- Risk Management begins with Risk Awareness
- Examples
 - Sickness of a skilled resource
 - Delivery fail for a supplier
 - Poor performance of a deliverable



Risk Management Process

- Risk Identification
- Risk Assessment
- Risk Evaluation
- Risk Management

Risk Management Process (MSDN)



Risk Identification [1]

- Risk Identification determines which risks might affect the project and documents their characteristics
- Participants in risk identification activities can include the following :
 - Project manager
 - Project team members
 - Risk management team
 - Customers
 - End users

Risk Identification [2]

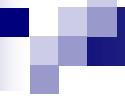
- Risk Identification is an iterative process
- New risks may become known as the project progresses through its life cycle

Risk Identification [3]

- Identify activities that may be affected by unexpected events
- Risks can be identified in a number of ways
 - General Risks – can happen to any task, resource or project
 - Risk specific to the project
 - Risks applicable to one task
 - Resource specific risks

Risk Identification [4]

- Review the schedule with experienced people
 - Refer to previous projects files and archives
 - Brainstorm likely events that may occur in your project
 - Perform SWOT analysis (Strength, Weaknesses, Opportunities, Threats)



Types of risks (from RUP)

- Resource Risks
- Business Risks
- Technical Risks
- Schedule Risks

Resource Risks (1)

■ Organization

- Is there sufficient commitment to this project (including management, testers, QA, and other external but involved parties)?
- Is this the largest project this organization has ever attempted?
- Is there a well-defined process for software engineering?

Resource Risks (2)

■ Funding

- Is the funding in place to complete project?
- Has funding been allocated for training and mentoring?
- Are there budget limitations such that the system must be delivered at a fixed cost or be subject to cancellation?
- Are cost estimates accurate?

Resource Risks (3)

■ People

- Are enough people available?
- Do they have appropriate skills and experience?
- Have they worked together before?
- Do they believe the project can succeed?
- Are user representatives available for reviews?
- Are domain experts available?

Resource Risks (4)

■ Time

- Is the schedule realistic?
- Can functionality be scope-managed to meet schedules?
- How critical is the delivery date?
- Is there time to "do it right"?

Business Risks

- What if a competitor reaches the market first?
- What if project funding is jeopardized (the other way to look at this is to ask "what can assure adequate funding")?
- Is the projected value of the system greater than the projected cost? (be sure to account for the time-value of money and the cost of capital).
- What if contracts cannot be made with key suppliers?

Technical Risks (1)

■ Scope risks

- Can success be measured?
- Is there agreement on how to measure success?
- Are the requirements fairly stable and well understood?
- Is the project scope firm or does the scope keep expanding?
- Are the project development time scales short and inflexible?

Technical Risks (2)

- **Technological risks**
 - Has the technology been proven?
 - Are reuse objectives reasonable?
 - An artifact must be used once before it can be re-used.
 - It may take several releases of a component before it is stable enough to reuse without significant changes.
 - Are the transaction volumes in the requirements reasonable?
 - Are the transaction rate estimates credible? Are they too optimistic?
 - Are the data volumes reasonable? Can they be held on currently available mainframes, or, if the requirements lead you to believe a workstation or departmental system will be part of the design, can the data reasonably be held there?
 - Are there unusual or challenging technical requirements that require the project team to tackle problems with which they are unfamiliar?
 - Is success dependent on new or untried products, services or technologies, new or unproven hardware, software, or techniques?
 - Are there external dependencies on interfaces to other systems, including those outside the enterprise? Do the required interfaces exist or must they be created?
 - Are there availability and security requirements which are extremely inflexible (for example, "the system must never fail")?
 - Are the users of the system inexperienced with the type of system being developed?
 - Is there increased risk due to the size or complexity of the application or the newness of the technology?
 - Is there a requirement for national language support?
 - Is it possible to design, implement, and run this system? Some systems are just too huge or complex to ever work properly.

Technical Risks (3)

- **External dependency risk**
 - Does the project depend on other (parallel) development projects?
 - Is success dependent on off the shelf products or externally-developed components?
 - Is success dependent on the successful integration of development tools (design tools, compilers, etc.), implementation technologies (operating systems, databases, inter-process communication mechanisms etc.). Do you have a back-up plan for delivering the project without these technologies?

Checklist [1]

- Key member of staff on critical path
- Key skill on long-duration task
- Project Manager on critical path
- External dependencies without tight contract
- Reliance on leading-edge technologies
- Many internal dependencies with conflicting priorities

Checklist [2]

- Ramping up resources
- Assumptions about availability of people and equipment
- Resource absences (Vacation time)
- Site shutdown for maintenance
- Health and safety requirements and certifications

Checklist [3]

- Identify risks that translate to project slippage
- Identify near critical path that could become critical
- Identify slack for non critical path
- Identify risks on critical path that translate to slippage

Risk Identification Output

- The outputs from Risk Identification are typically contained in a document that can be called a **risk register** and contains :
 - List of identified risks
 - Likely consequences
 - List of potential responses
 - Root causes of risk
 - Updated risk categories

Risk Register

	A	B	C	D	E	F	G	H	I
1	ADC APPROVED RISK REGISTER - Up-dated January 2007								
2	Ref	Type	Hazard/Threat/Vulnerability	Likely consequence	Risk rating	Lead Officer	Division	Planned action - list individual actions	Progress to date Jan 2007
3	1	REG	Application of policies and procedures	Tasks/functions performed incorrectly; inefficiencies; variable standards of service delivery; local variations are established; dangerous practices become commonplace; incidents occur; unable to effectively defend claims	8B	I. Lowrie	CMT	Training. Promotion/publicity. Procedure notes. Monitoring	Included in staff induction process to raise awareness of corporate documents with new employees. On going responsibility lies with divisional managers for the education and compliance of their staff.
4	2	PEO	Senior officer capacity	Operational issues not tackled; strategic issues not tackled; issues fester/store up; issues blow up; impact on service delivery; standards of service delivery fall; no excess capacity to deal with new initiatives/legislation; increased pressure on staff; long hours; staff leave; public complaints; media attention; reputation damage to Council	16	I. Lowrie	CMT	Subsumed within report on re-structuring 2005. To be monitored through staff survey results, sickness records, CPT and CMT. Links to risk 12	Continue to monitor. Subsume within final considerations of whether to create a single officer structure and merge services with WBC in report March 2007
5	3	STRAT	Economic development /inward investment	Local economy suffers; reputation of area declines; area not seen as attractive to employers; other businesses choose not to locate in area; adverse impact on businesses; economic growth not achieved; economic regeneration agenda not fulfilled; area fails to prosper; neighbouring areas prosper; Council criticised for not being pro-active; negative local publicity; image of Council damaged	4	P. Davies	CMT	Review regeneration strategy. Rename "Revitalising Adur". Report to PSC Feb 2006	Employment study completed. AIF funded projects started. Enterprise gateway opened. Inward Investors in airport.

Risk Assessment Template

Risk Assessment

- Identify Probability and Impact of each risk
- Key question: what event will have an influence on the project?
- Probability: How likely is it to happen?
- Impact: What will be the consequences?

Qualitative Risk Assessment

- includes methods for prioritizing the identified risks for further action, such as :
 - Quantitative Risk Analysis
 - Risk Response Planning
- determines the priority of identified risks using their probability of occurring and the corresponding impact on project objectives

Qualitative Assessment Tools

■ Probability and Impact Matrix

- Ratings are assigned to risks based on their assessed probability and impact
- Probability and impact matrix => rating the risks as low, moderate, or high priority

Risk matrix example

	1 Low	3 Medium	7 High	9 Critical	
Cost 2	Insignificant	<5%	<10%	>10%	
Time 3	Insignificant	<10 days	<25 days	>25 days	
Scope 5	Insignificant	No business impact	Some business impact	Definite business impact	
Quality 2	Insignificant	No anticipated problems	Potential for problems	Definite problems	

Example – Inadequate training

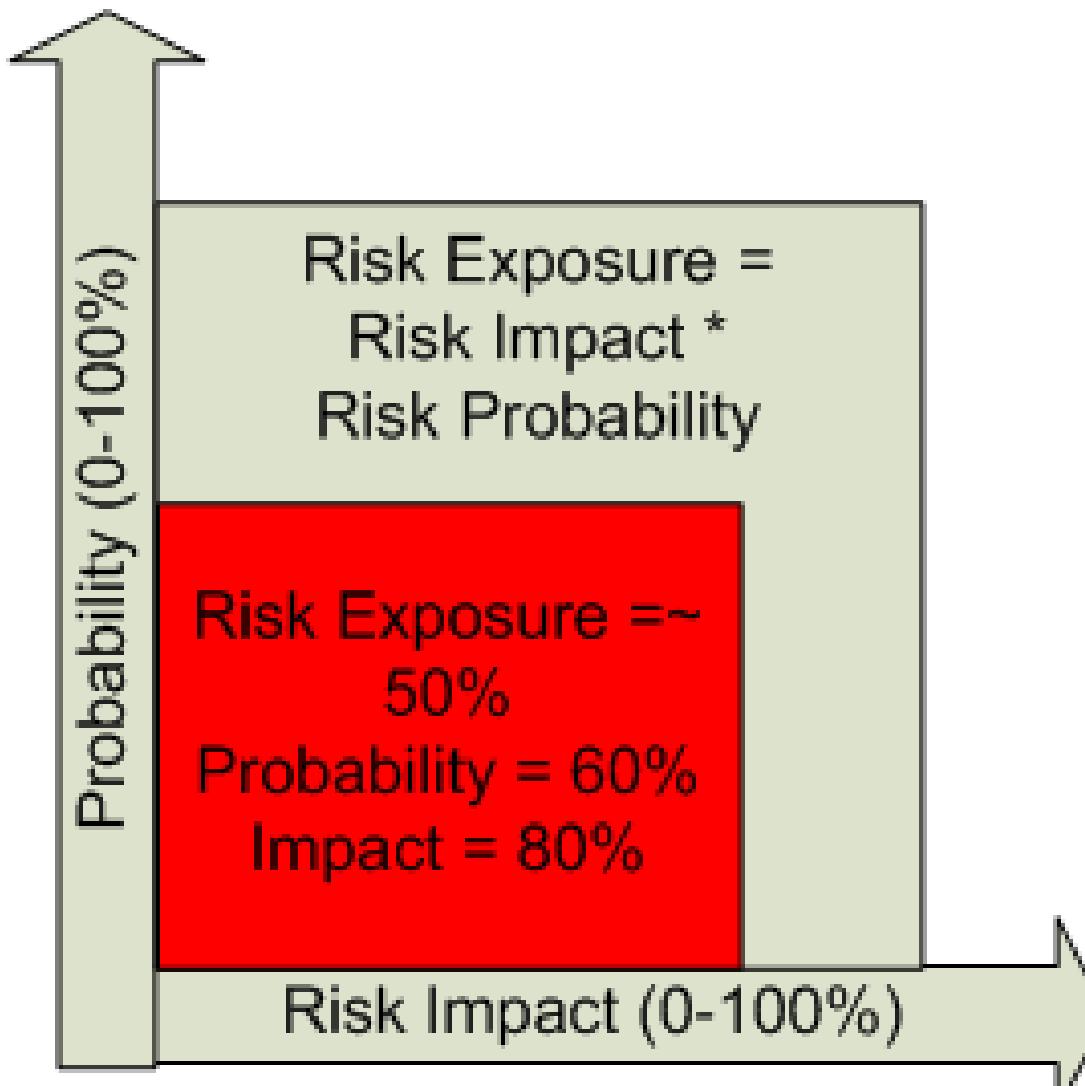
	Low 1	Medium 3	High 7	Critical 9		
Cost 2	Insignificant	<5%	6	<10%	>10%	
Time 3	Insignificant	<10 days		<25 days	>25 days	27
Scope 5	Insignificant 5	No business impact		Some business impact	Definite business impact	
Quality 2	Insignificant	No anticipated problems	Potential for problems	14	Definite problems	

$$6+27+5+14 = 52 \Rightarrow \text{Medium impact}$$

Evaluation example

- Critical > 70
- Medium > 50
- Low < 50

Risk Exposure



Quantitative Risk Assessment

[1]

- Is performed on risks that have been prioritized as High-Risks by the Qualitative Risk Analysis process
- Analyzes the effect of those risk events and assigns a numerical rating to those risks

Quantitative Risk Assessment

[2]

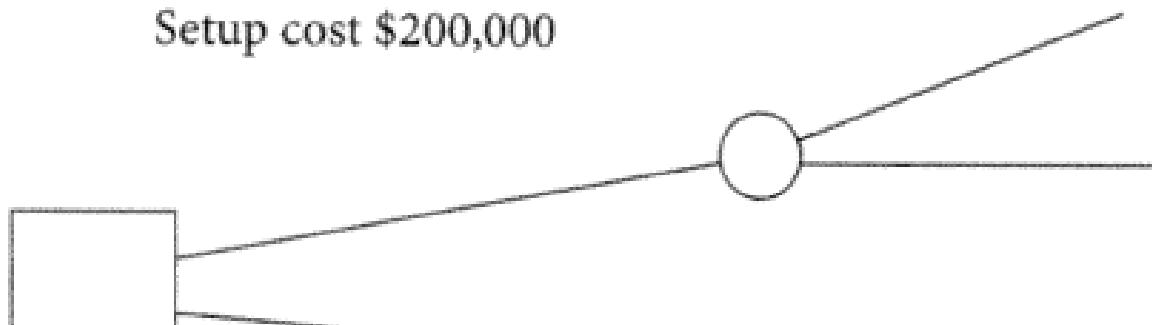
- Uses techniques such as Monte Carlo simulation and decision tree analysis to:
 - Quantify the possible outcomes for the project and their probabilities
 - Identify risks requiring the most attention by quantifying their relative contribution to overall project risk

Exercise

- Prototyping is worthwhile on the project?

Prototype:

Setup cost \$200,000



Do not prototype:

Setup cost \$0

$$200.000 + 35/100 * 120.000 = 242.000$$

$$200.000 + 35/100 * 120.000 = 242.000$$

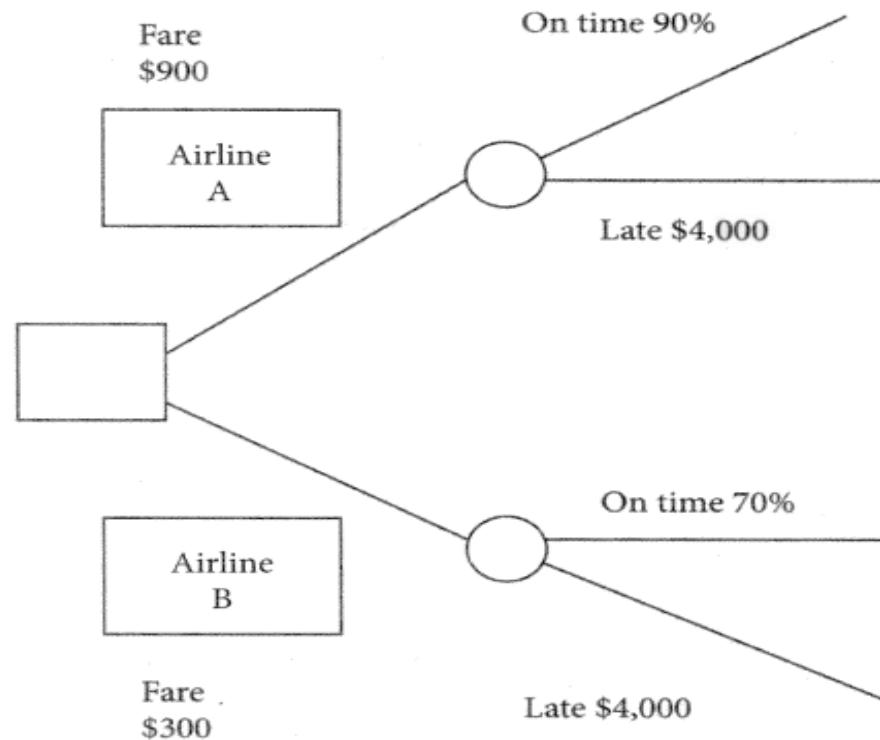
Failure: 35% probability
and \$120,000 impact

Pass: No impact

Failure: 70% probability and
\$450,000 impact

Pass: No impact

Another exercise



Airline A

$$(10\% \times \$4,000) + (90\% \times \$0) + \$900 \\ \$400 + \$0 + \$900 = \$1,300$$

Airline B

$$(30\% \times \$4,000) + (70\% \times \$0) + \$300 \\ \$1,200 + \$0 + \$300 = \$1,500$$

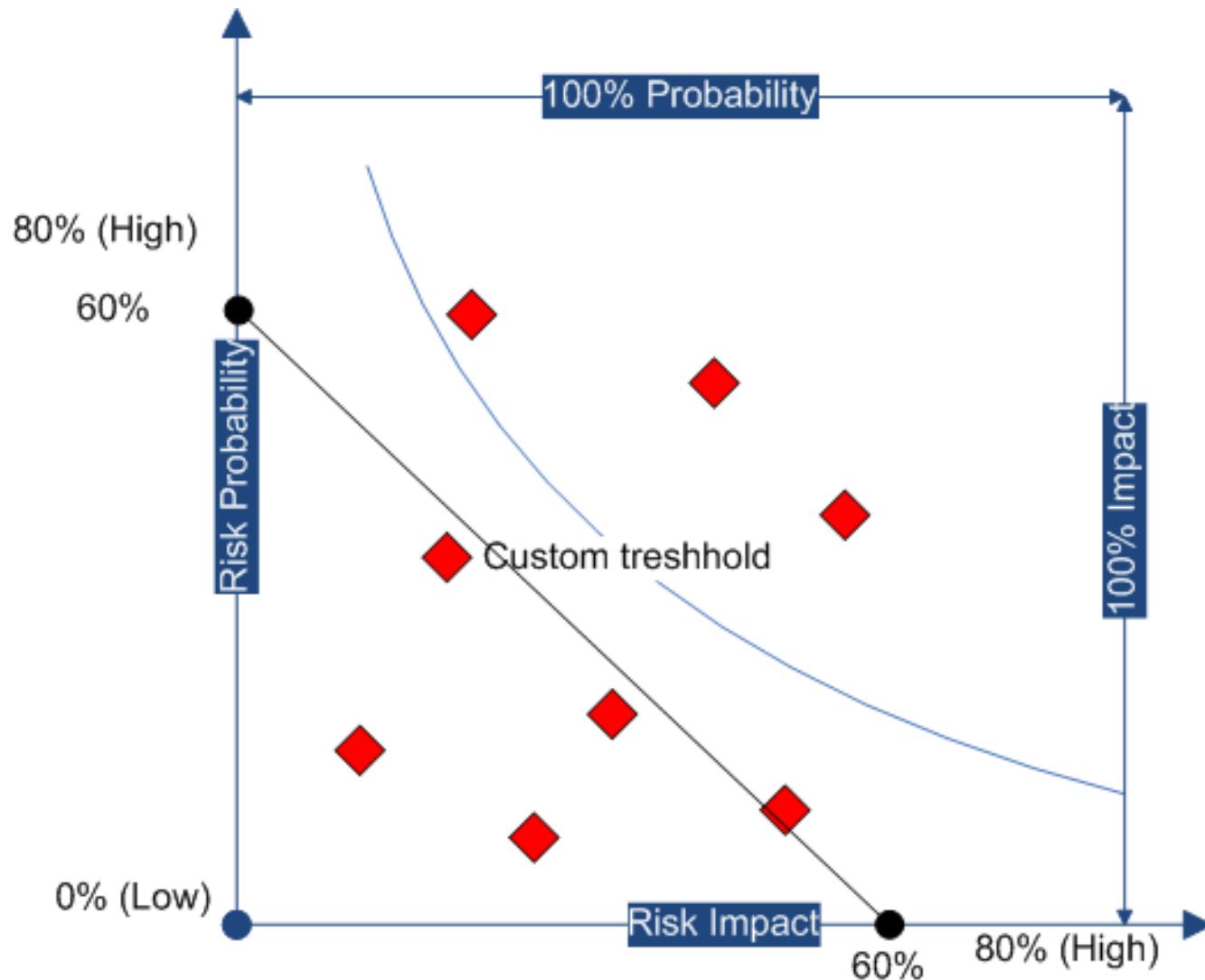
Risk Evaluation [1]

- Risks can be placed on a graph according to probability and impact
- Decide the zone on the graph that contains unacceptable risks:
 - Depends on organization
 - Depends on domain
 - Depends on project

Risk Evaluation [2]

- To move each risk out of that zone:
 - Reduce the probability
 - Reduce the impact
 - => Reduce exposure
- These actions are recorded in the risk register:
 - Mitigation (**what** will be done to prevent?)
 - Triggers (**when** will it be done?)
 - Responsible (**who** will do it?)

Risk Management Chart



Another Example

Probability and Impact Matrix

Probability	Threats					Opportunities				
	0.90	0.05	0.09	0.18	0.36	0.72	0.72	0.36	0.18	0.09
0.70	0.04	0.07	0.14	0.28	0.56	0.56	0.28	0.14	0.07	0.04
0.50	0.03	0.05	0.10	0.20	0.40	0.40	0.20	0.10	0.05	0.03
0.30	0.02	0.03	0.06	0.12	0.24	0.24	0.12	0.06	0.03	0.02
0.10	0.01	0.01	0.02	0.04	0.08	0.08	0.04	0.02	0.01	0.01
	0.05	0.10	0.20	0.40	0.80	0.80	0.40	0.20	0.10	0.05

- High Risk : dark gray (largest numbers)
- Moderate risk : light gray (in-between numbers)
- Low risk : medium gray (smallest numbers)

Risk Metrics

■ Measurements

- E_P - Total Project Exposure
- E_i – Exposure for Risk i
- I_i – Impact for Risk i
- P_i – Probability for Risk i

Risk Metrics Trend

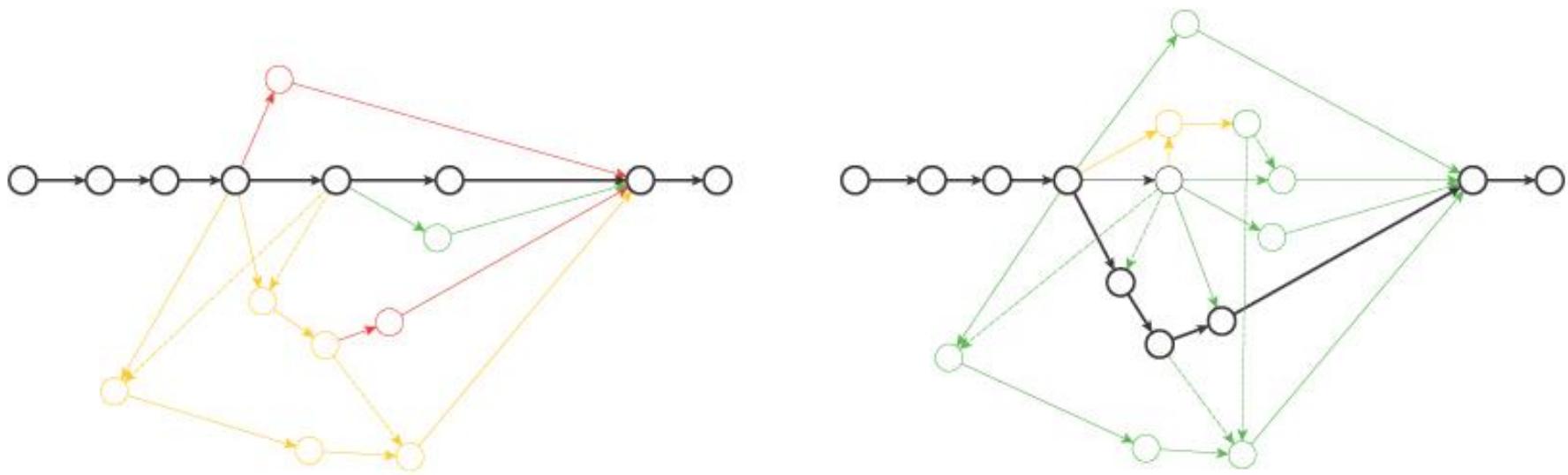
- Risk Exposure Metric (should $\rightarrow 0$)

$$E_P = \sum_{i=1}^n E_i = \sum_{i=1}^n P_i * I_i$$

- Risk Exposure Metric Trend (should <0)

$$\overline{E}_P = \frac{dE_p}{dt}$$

Activity risk modeling and quantification



Activity classification

- How do we decide the color of an activity?
- Thresholds
- Ex.
 - $\text{Float} > 25 \Rightarrow \text{green}$
 - $15 < \text{Float} \leq 25 \Rightarrow \text{yellow}$
 - $0 < \text{Float} \leq 15 \Rightarrow \text{red}$

Criticality Risk model

$$\text{Risk} = \frac{(W_C * N_C + W_G * N_G + W_Y * N_Y + W_R * N_R)}{W_C * N}$$

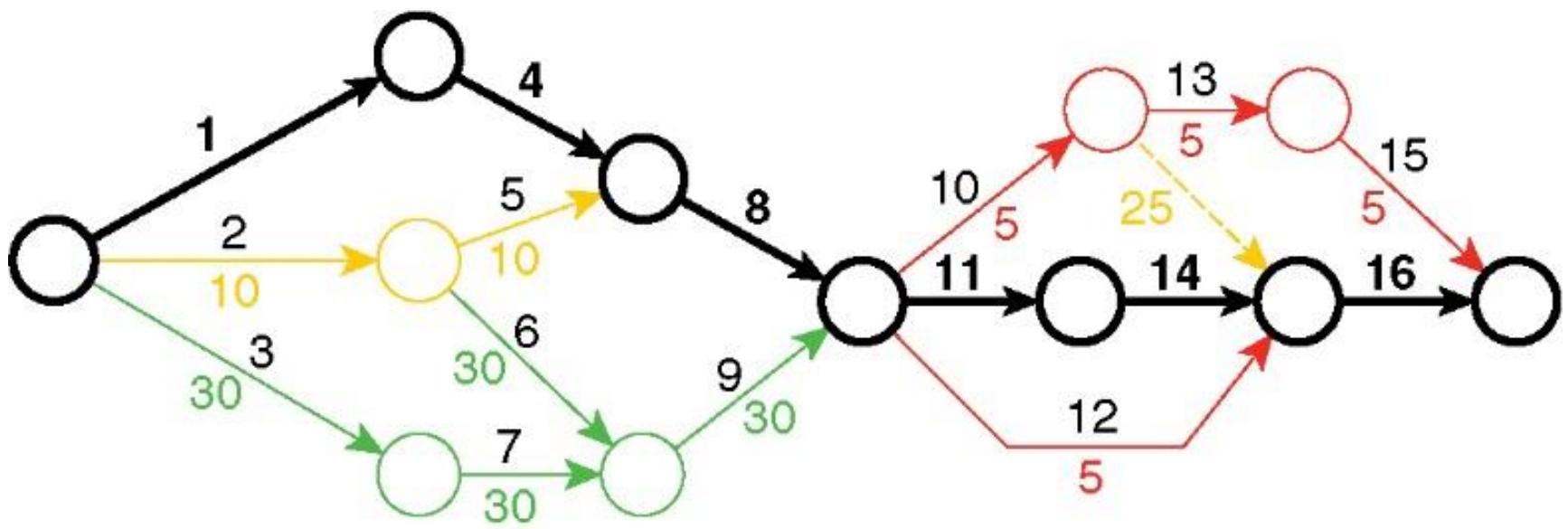
where

$W_{C,G,Y,R}$ – weight of critical/green/yellow/red activities

$N_{C,G,Y,R}$ – number of critical/green/yellow/red activities

$N = N_C + N_G + N_Y + N_R$ (total number of activities)

Example



$$W_C = 4, W_R = 3, W_Y = 2, W_G = 1$$

$$\text{Risk} = (4*6 + 1*4 + 2*3 + 3*4) / 4*17 =$$

What happens if

- All activities in the network are critical?
 - Risk = $4^*N/4^*N = 1$ (maximum!)

- All activities in the network are green?
 - Risk = W_G/W_C (impossible!)

What are appropriate weights?

- 1, 2, 3, 4
- 21, 22, 23, 24?

Hint: minimum risk = $W_G/W_C = 21/24 \sim 0.875$
(too high!)

What are appropriate weights?

- Good practice: Fibonacci series
 - $\text{Fib}_i = \text{Fib}_{i-1} + \text{Fib}_{i-2}$
 - $\text{Fib}_i = \varphi^* \text{Fib}_{i-1}$, for large i
 - $\varphi = 1.618\dots$ (golden ratio)
- $W_Y = \varphi^* W_G, W_R = \varphi^{2*} W_G, W_C = \varphi^{3*} W_G$

$$\text{Risk} = (\varphi^3 N_C + \varphi^2 N_R + \varphi N_Y + N_G) / \varphi^3 N$$

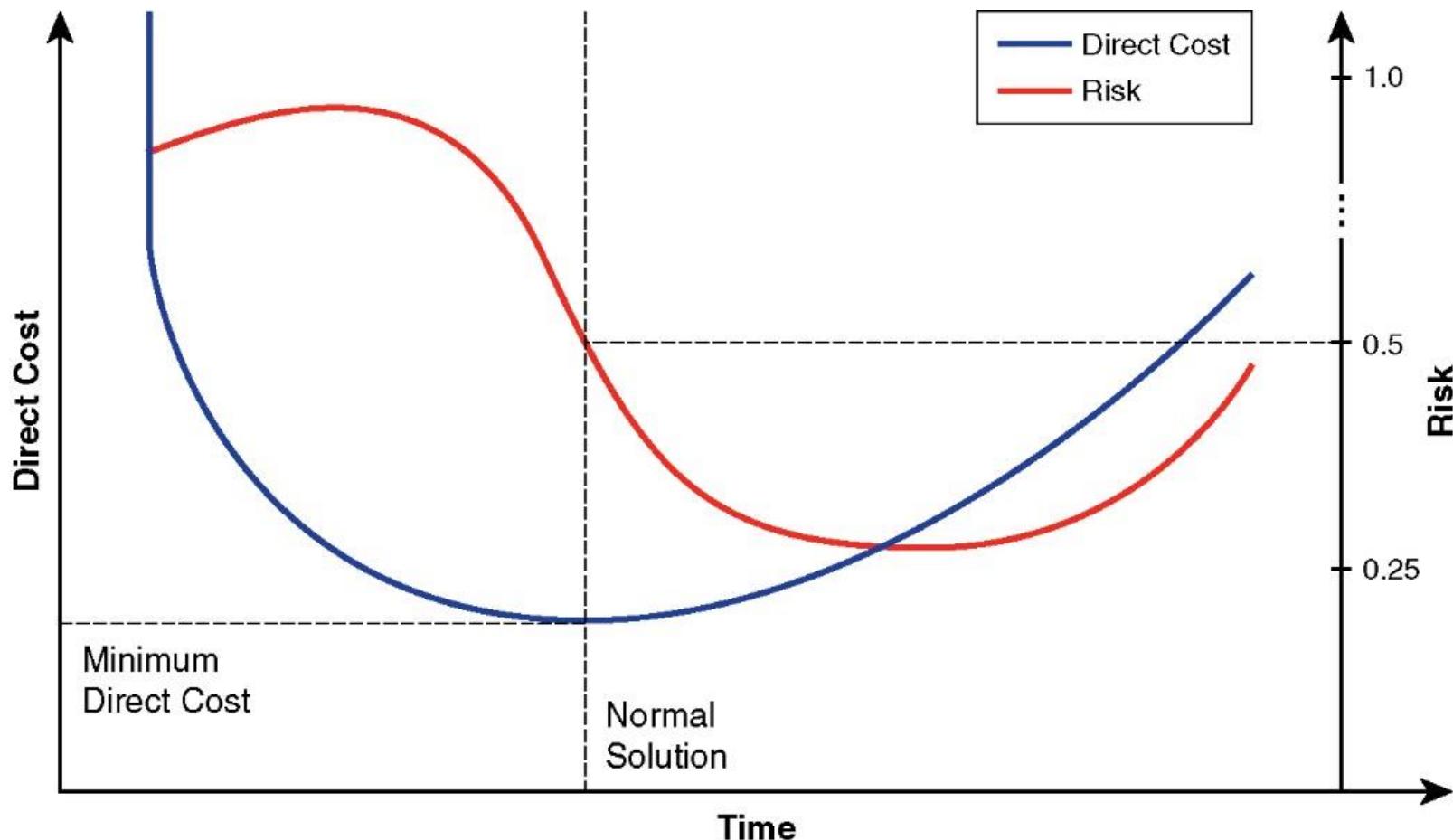
Activity risk model

- Instead of weights, use floats

Risk = $1 - (F_1 + F_2 + \dots + F_N) / M * N$ where

- F_i = float of activity i
- $M = \text{Max}(F_1, F_2, \dots, F_N)$
- N = number of activities

Risk and direct cost models



Risk Response Strategies (for threats)

- Avoidance
- Reduction (Mitigate)
- Transfer
- Acceptance => consider contingency plans!!

Risk Response Strategies (for opportunities)

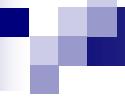
- Exploit (opposite to Avoidance)
- Enhance (opposite to Reduce)
- Share (opposite to Transfer)
- Acceptance

Exercise

Description	Strategy
Remove a work package from the project	Avoid
Assign a team member to visit the suppliers' facility	Mitigate impact
Move a work package to a date when a more experienced resource is available	Exploit
Begin negotiation for an equipment early to insure a better price	Enhance impact
Notify management that there could be a cost increase due to a risk because no action was taken to	Accept
Provide a team member having limited experience with additional training	Mitigate probability
Outsource difficult work to a more experienced company	Transfer
Prototype a risky piece of equipment	Mitigate probability
Ask the client to handle some of the work	Transfer

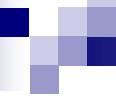
Risk Assessment Document

- Risk **statements**, which capture the nature of each risk
- Risk **probability**, which describes the likelihood of the occurrence of the risk
- Risk **severity**, which specifies the impact of the risk
- Risk **exposure**, which specifies the overall threat of the risk
- **Mitigation plans**, which describe the efforts for preventing or minimizing the risk
- **Contingency plans and triggers**, which specify the steps that you need to take when a risk occurs and when to take those steps
- Risk **ownership**, which specifies the name of the team member who is responsible for monitoring the risk on a regular basis



Risk Management Tool

- Risk Template Tool

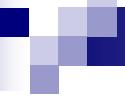


Risk indicators

- Total impact of top ten risks
- Top ten risks mitigation costs
- Top ten risk variation
- Top ten risks impact variation

Risk management key points

- Iterative development enables improved risk management
- Risks should be monitored
 - Continuously
 - Proactively
- Risk management should be planned and the effort for it estimated (if the effort exceeds the risk, it is not risk)



Risk management in Agile

- Has the adoption of Agile techniques magically erased risk from software projects?
- Can Agile and lean techniques be leveraged to make managing risk part of the day-to-day activities of teams while reducing overhead?

Recognition: Risk In Agile Is Different

*“A great deal of **explicit** risk management becomes unnecessary when a project uses an agile approach”*

Mike Cohn, Mountain Goat Software

Approach

Risk	Agile Approach
Mitigating Schedule Flaw	Scrum provides feedback loops to mitigate invalid estimates. Teams update the release plan at the end of every.
Mitigating Specification Breakdown	A scrum delivery team will work collaboratively with the product owner to ensure alignment between what is requested and how it can be delivered.
Mitigating Scope Creep	The product owner will evaluate the new backlog items and decide what action to take: add, delete, trade-out in priority with other product backlog items.
Mitigating Personnel Loss	Self-organizing teams focus problems impacting work resulting in higher morale.
Mitigating Productivity Variation	Agile teams address the performance at the end of every sprint as part of the retrospective.

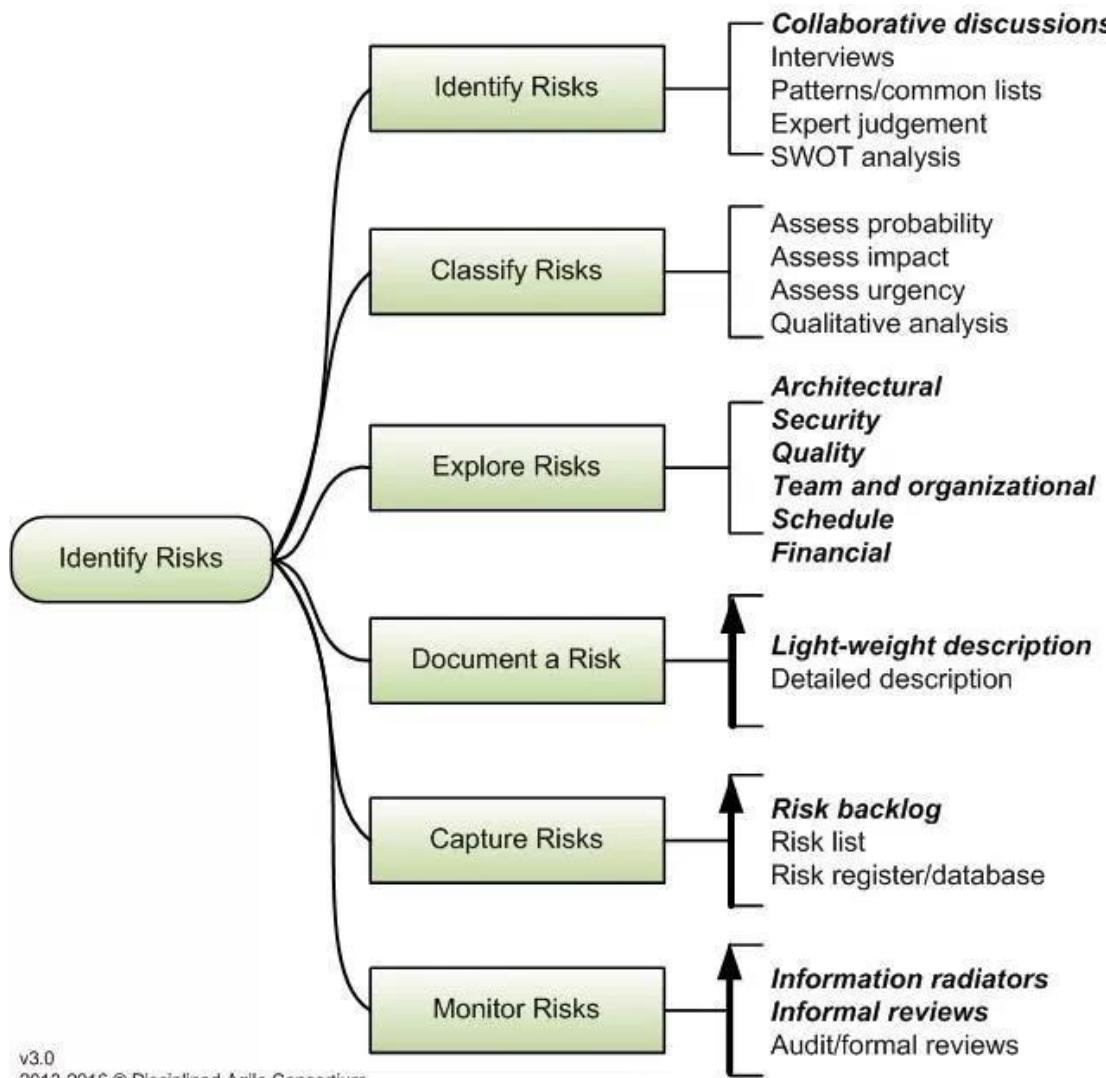
Agile risk management

- Identify knowable risks when generating the initial backlog.
- Build mitigation for common risks into the definition of done.
- Generate stories for less common risks and add them to the projects backlog.
- Review risks when grooming stories
- Carve out time during planning to identify emerging risks

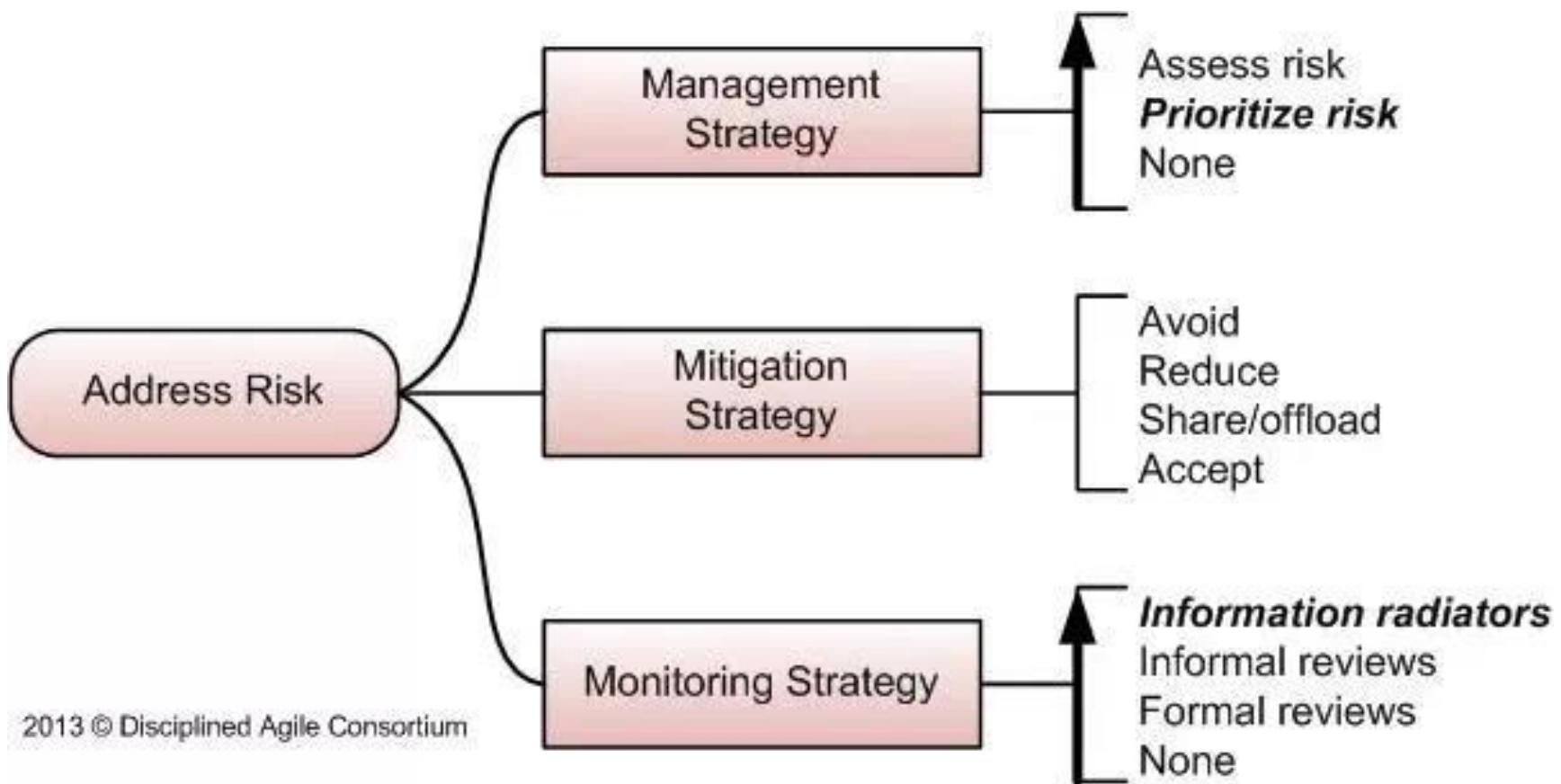
Risk recognition

- Carve out time when you are developing the backlog and ask as diverse a group possible to identify the potential problems.
- Form a small team to interview stakeholders that were not part of the planning exercise.
- Gather risk data through surveys when the program stakeholders are geographically diverse.
- Interview customers or potential customers.
- Periodically discuss risks either as an agenda item or as a follow-on to standard meetings.

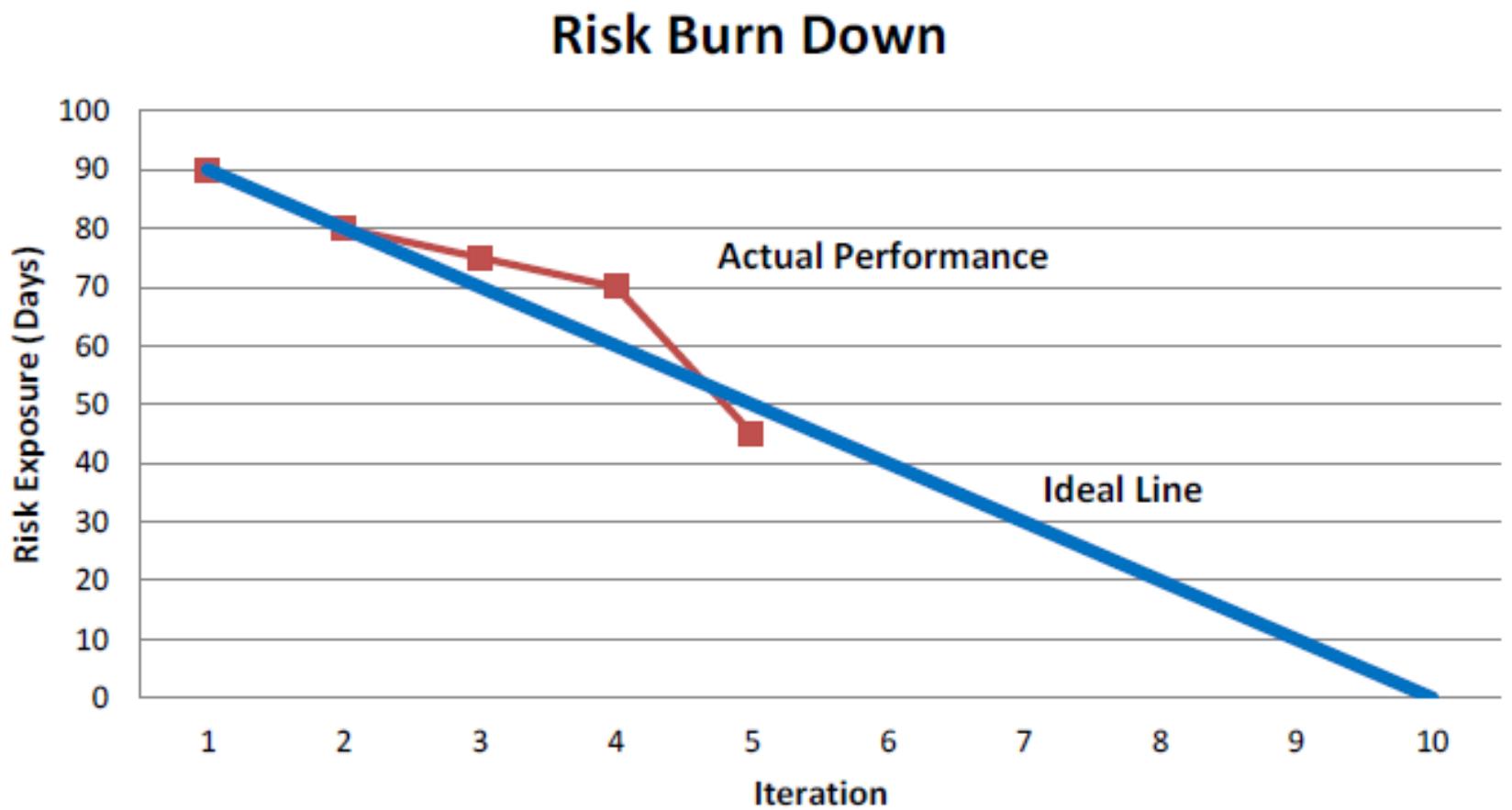
Risk identification in DAD

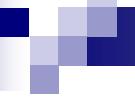


Risk management in DAD



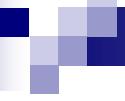
Measuring Risk





Lessons learned

- Risk management become a series of conversations not a series of documents.
- Each build is assessed, issues identified and the backlog of tasks is reviewed and prioritized and the most important tasks, issues and risk mitigation are scheduled for the next sprint.

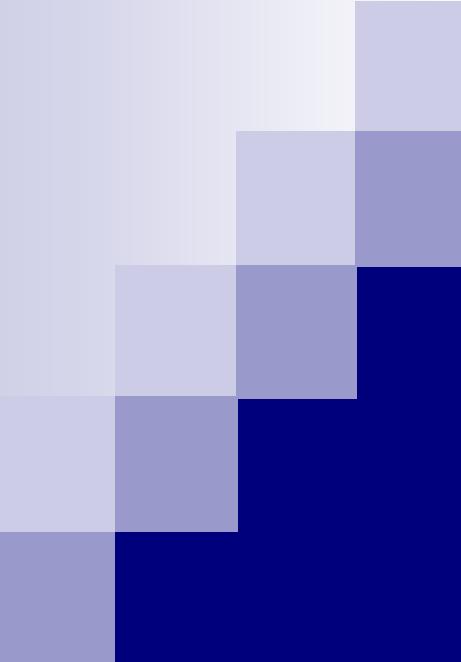


Quiz time

- Please switch over to Moodle

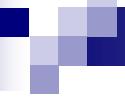
Discussion topic for next time

- Give an example of a specific risk, possible specific responses (and their type). Simulate a quantitative analysis.



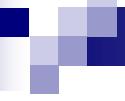
Monitoring and Control

Lecture 9



...recap::Risk Management

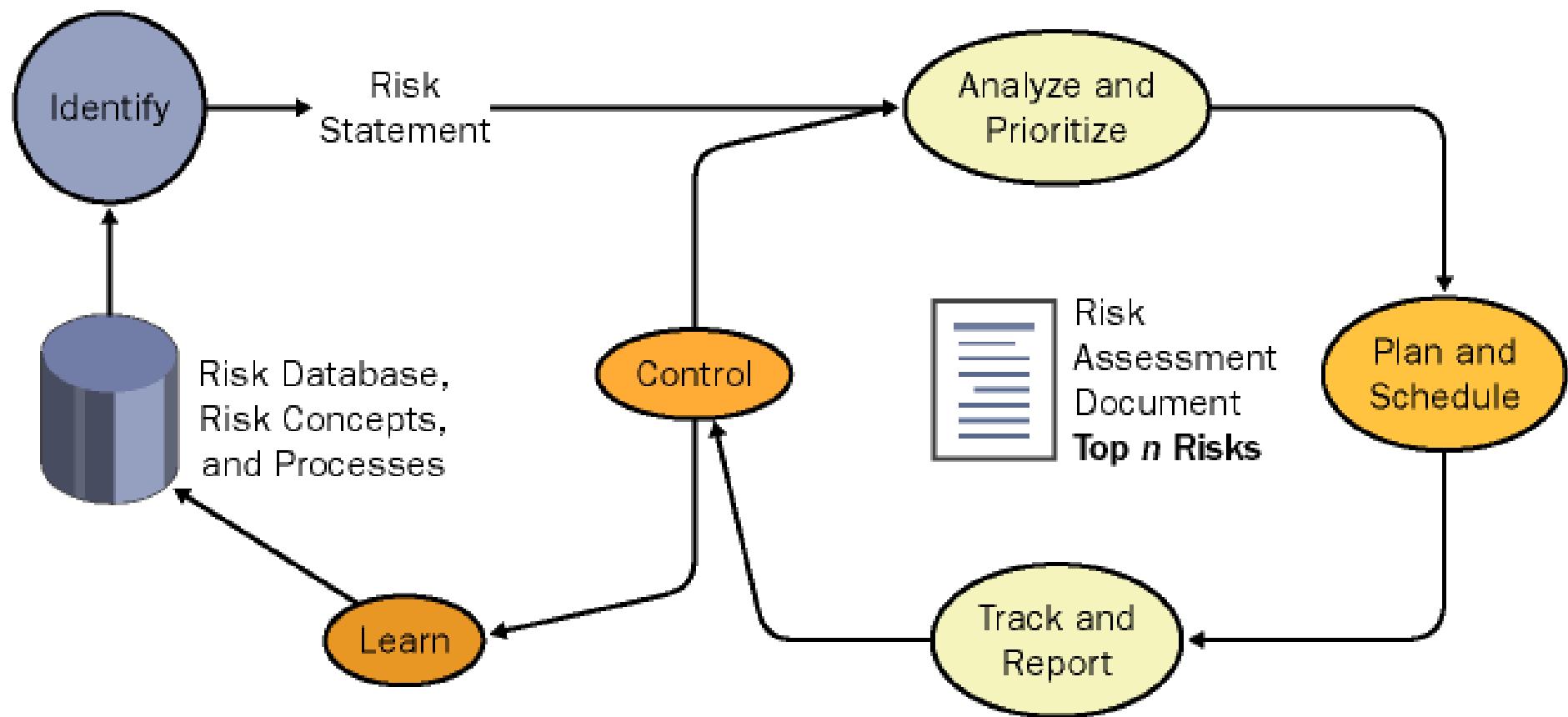
- Risk Management begins with Risk Awareness



...recap:: RM Process

- Risk Identification
- Risk Assessment
- Risk Evaluation
- Risk Management

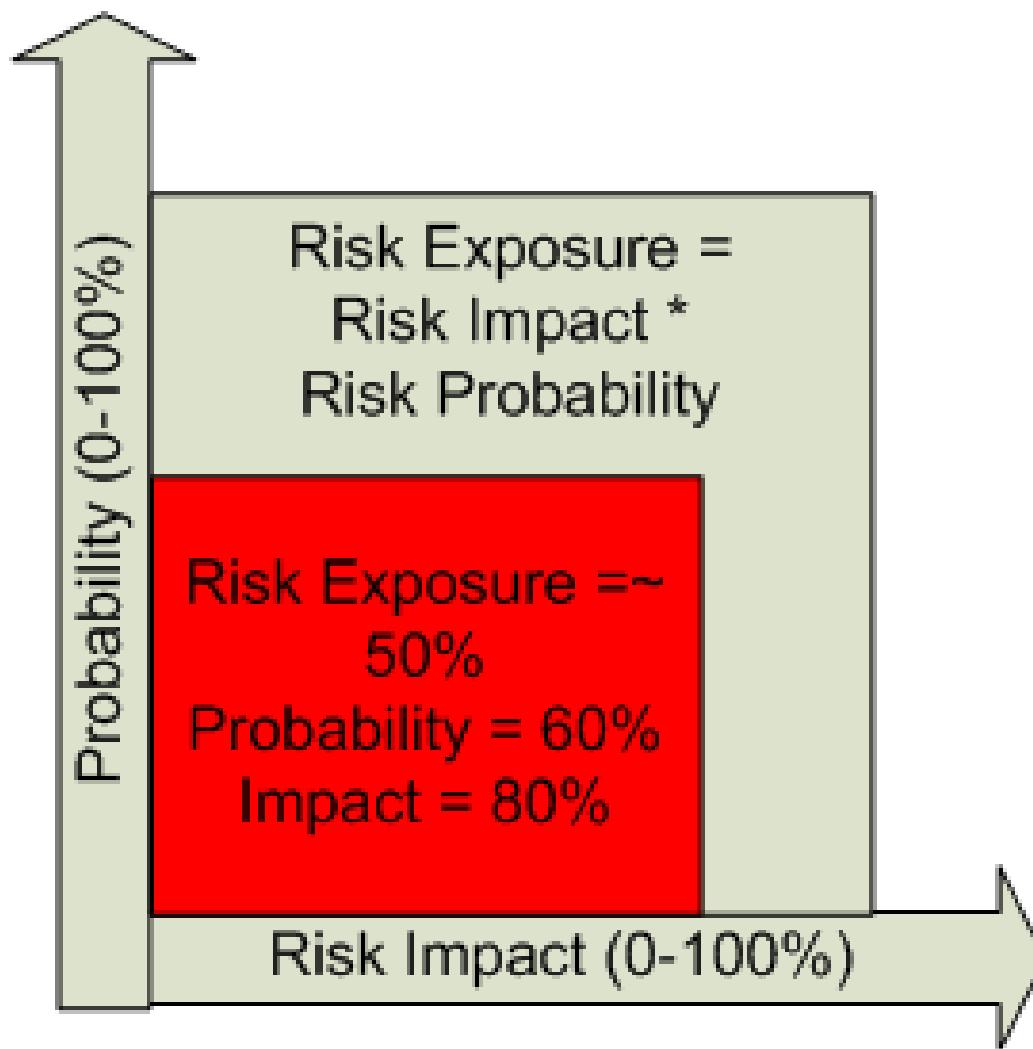
...recap:: RM Process



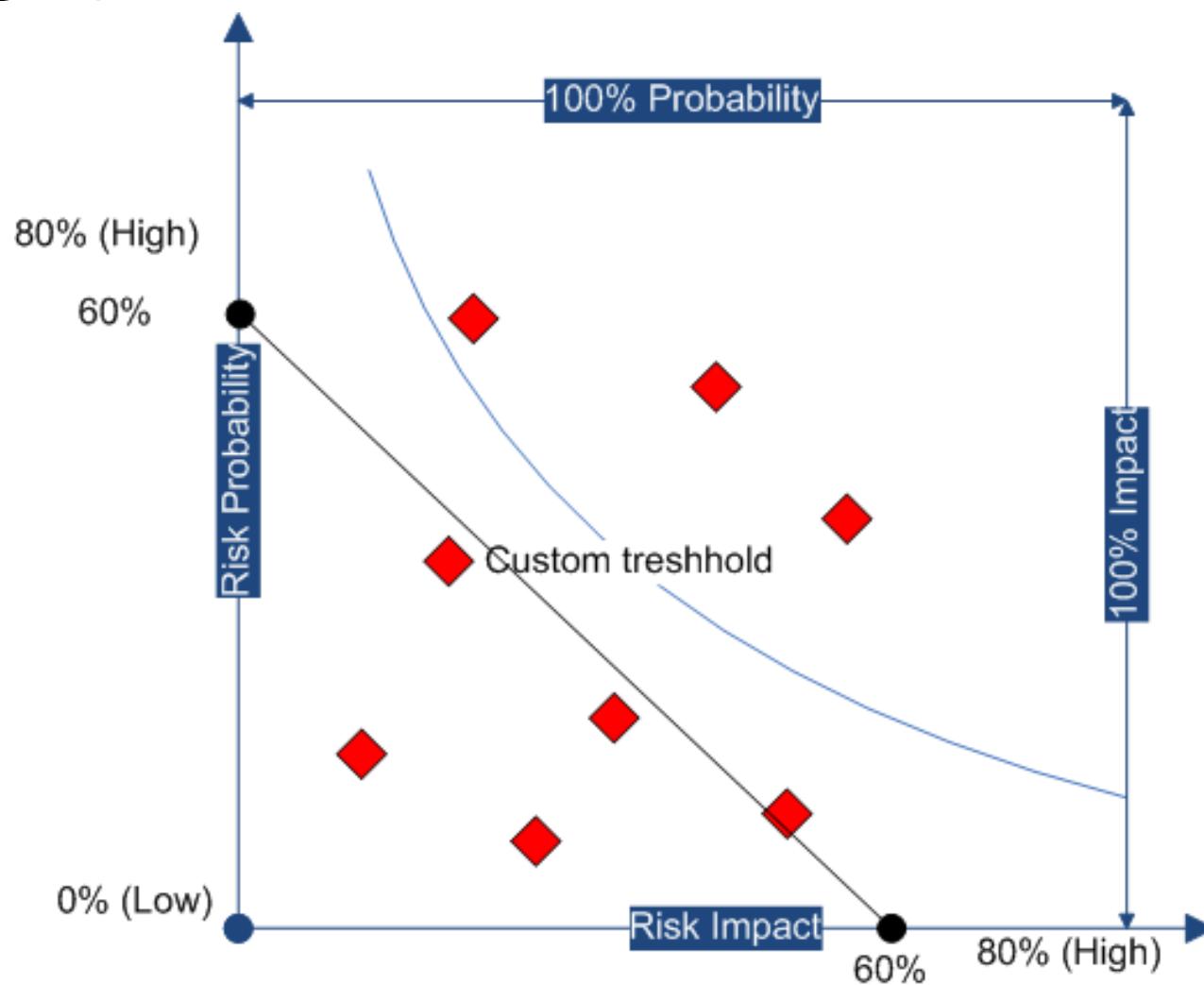
...recap::RM Risk Register

	A	B	C	D	E	F	G	H	I
1	ADC APPROVED RISK REGISTER - Up-dated January 2007								
2	Ref	Type	Hazard/Threat/Vulnerability	Likely consequence	Risk rating	Lead Officer	Division	Planned action - list individual actions	Progress to date Jan 2007
3	1	REG	Application of policies and procedures	Tasks/functions performed incorrectly; inefficiencies; variable standards of service delivery; local variations are established; dangerous practices become commonplace; incidents occur; unable to effectively defend claims	8B	I. Lowrie	CMT	Training. Promotion/publicity. Procedure notes. Monitoring	Included in staff induction process to raise awareness of corporate documents with new employees. On going responsibility lies with divisional managers for the education and compliance of their staff.
4	2	PEO	Senior officer capacity	Operational issues not tackled; strategic issues not tackled; issues fester/store up; issues blow up; impact on service delivery; standards of service delivery fall; no excess capacity to deal with new initiatives/legislation; increased pressure on staff; long hours; staff leave; public complaints; media attention; reputation damage to Council	16	I. Lowrie	CMT	Subsumed within report on restructuring 2005. To be monitored through staff survey results, sickness records, CPT and CMT. Links to risk 12	Continue to monitor. Subsume within final considerations of whether to create a single officer structure and merge services with WBC in report March 2007
5	3	STRAT	Economic development /inward investment	Local economy suffers; reputation of area declines; area not seen as attractive to employers; other businesses choose not to locate in area; adverse impact on businesses; economic growth not achieved; economic regeneration agenda not fulfilled; area fails to prosper; neighbouring areas prosper; Council criticised for not being pro-active; negative local publicity; image of Council damaged	4	P. Davies	CMT	Review regeneration strategy. Rename "Revitalising Adur". Report to PSC Feb 2006	Employment study completed. AIF funded projects started. Enterprise gateway opened. Inward Investors in airport.

...recap: RM Risk Exposure

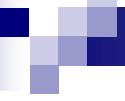


...recap: RM Risk Management Chart



...recap::RM Principles of Risk Management

- Proactive
- Continuous
 - Monitoring
 - Assessment
 - Decision making
- Expected result: Continuous risk reduction



...recap::Types of risks

- Resource Risks
- Business Risks
- Technical Risks
- Schedule Risks

...recap: Risk Response Strategies (-)

- Risk avoidance
- Risk reduction
- Risk transfer
- Risk acceptance

...recap::Risk Response Strategies (+)

- Exploit
- Share
- Enhance
- Risk acceptance

...recap::Risk Indicators

- Total impact of top ten risks
- Top ten risks mitigation costs
- Top ten risk variation
- Top ten risks impact variation

Today's topic

- Project Monitoring and Control

PMC topics for today

- Briefly describe monitoring purpose, deliverables, activities, measures
- Controlling processes, as described in PMBOK
- Some specific metrics examples

Definition

- The goal of Project Monitoring and Control (PMC) is to provide an **understanding of the project's progress** so that appropriate **corrective actions** can be taken when the project's performance deviates significantly from the plan.

Project Monitoring & Control

- Monitoring – collecting, recording, and reporting information concerning project performance that project manager and others wish to know
- Controlling – uses data from monitor activity to bring actual performance to planned performance

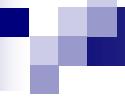
PMC

- **Why** do we monitor?
- **What** do we monitor?
- **When** do we monitor?
- **How** do we monitor?

Hawthorne effect: **We impact what we measure!**

Why do we monitor? Goals

- Provide project manager and development team with the following:
 - An accurate assessment of the **progress** to date
 - Insight into the **quality** of the evolving software product
 - A basis for **estimating cost and budgeting** with increased accuracy over time



What do we monitor?

- Men (human resources)
- Machines
- Materials
- Money
- Space
- Time
- Tasks
- Quality/Technical Performance

Purpose

- Ensure the team is making progress
- Overall objectives:
 - Track accomplishments and compare to planned values
 - Revise the project plan: accomplishments so far, remaining work (if needed)
 - Detect performance issues as early as possible, such that corrective action can be taken

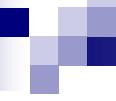
What do we obtain?

Inputs

- Time
- Money
- Resources
- Material Usage
- Tasks
- Quality/Technical Performance

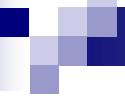
Outputs

- Progress
- Costs
- Job starts
- Job completion
- Engineering / Design changes
- Variation order (VO)



Deliverables

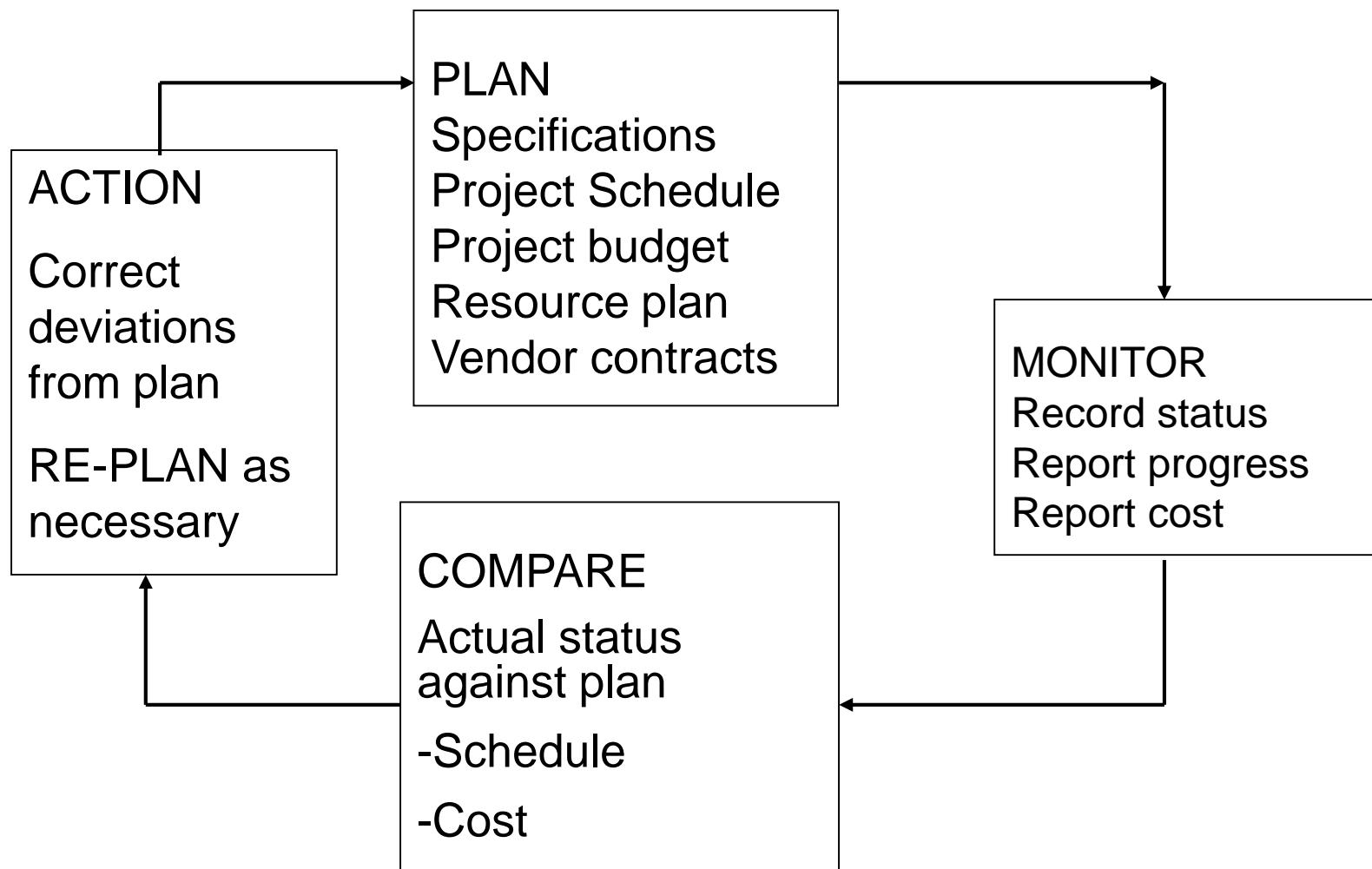
- **Written status reports**
- **Updates** to lists of action items, risks, problems, and issues
- **Updates to plan/schedule**, to reflect actual progress
- **Comparison** of actual vs. budgeted cost, cost/benefit analysis (EVA)
- **Audit/review** reports of the activities and work products under development



When do we monitor?

- Continuously
- Regularly
- Logically
- While there is still time to react
- As soon as possible
- At task completion
- At pre-planned decision points
(milestones)
- End of the project

Project Control Cycle



PMC – Activities (How?)

- Continuously monitor progress
 - Examine progress on all key dimensions of the project; goals likely to be met?
- Conduct team reviews
 - Communicate status (technical activities); plan for next activities of the project
- Conduct formal progress reviews
 - Monthly basis with senior management and key stakeholders

PMC – Activities (contd.)

■ Manage changes

- Identify, evaluate, prioritize, and control changes to the project

■ Revise the plan

- Significant changes need to be reviewed and agreed to by those who originally approved the plan

■ Conduct work product reviews

- Walkthroughs, technical reviews and inspections, based on quality goals

PMC – Example Measures

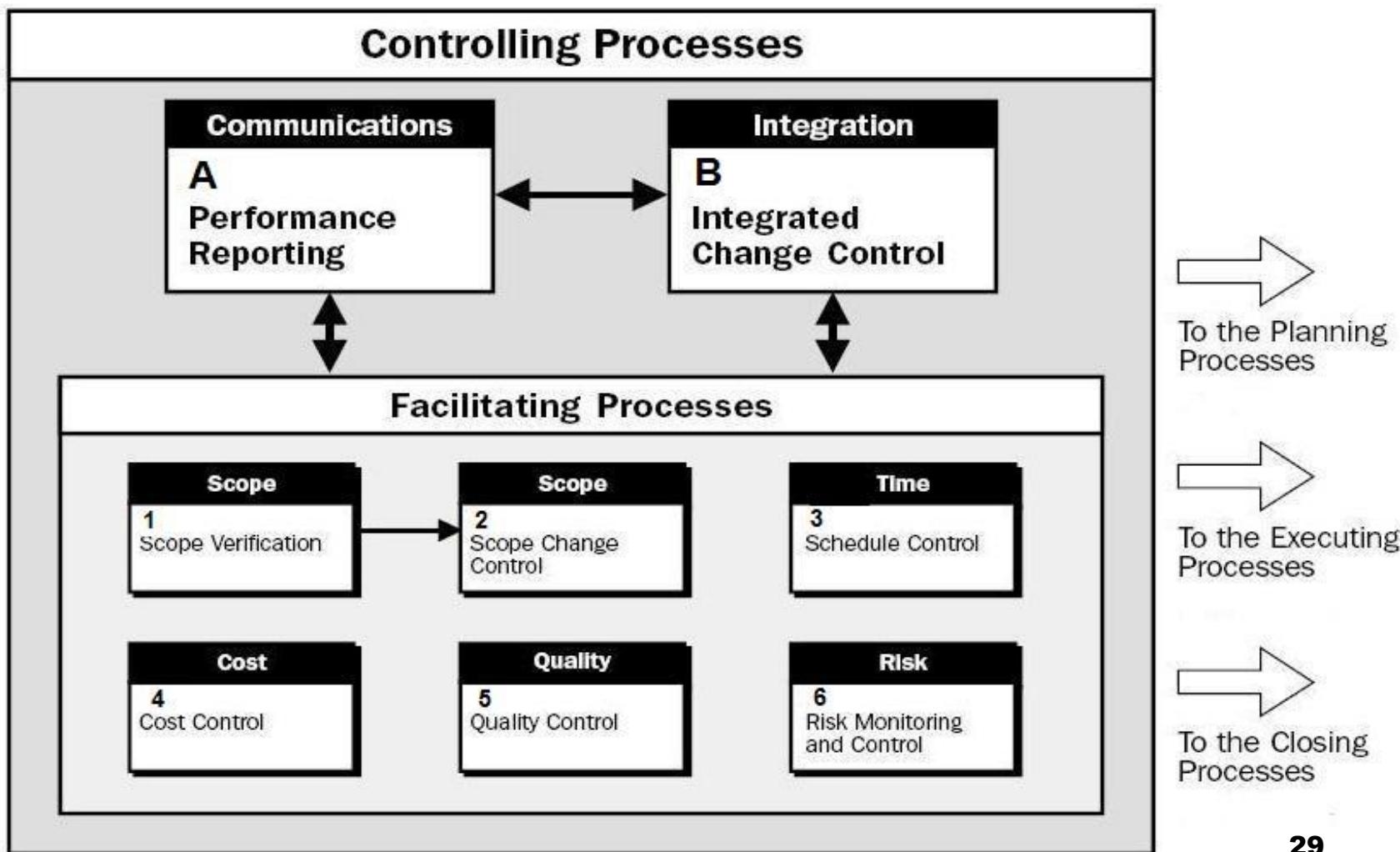
- Milestone attainment
 - Maintain the initial baseline, as well as the most recent update
 - Report achievement and variance to both

- Effort spent
 - Compare initial effort estimates for each major WBS element with actual effort spent

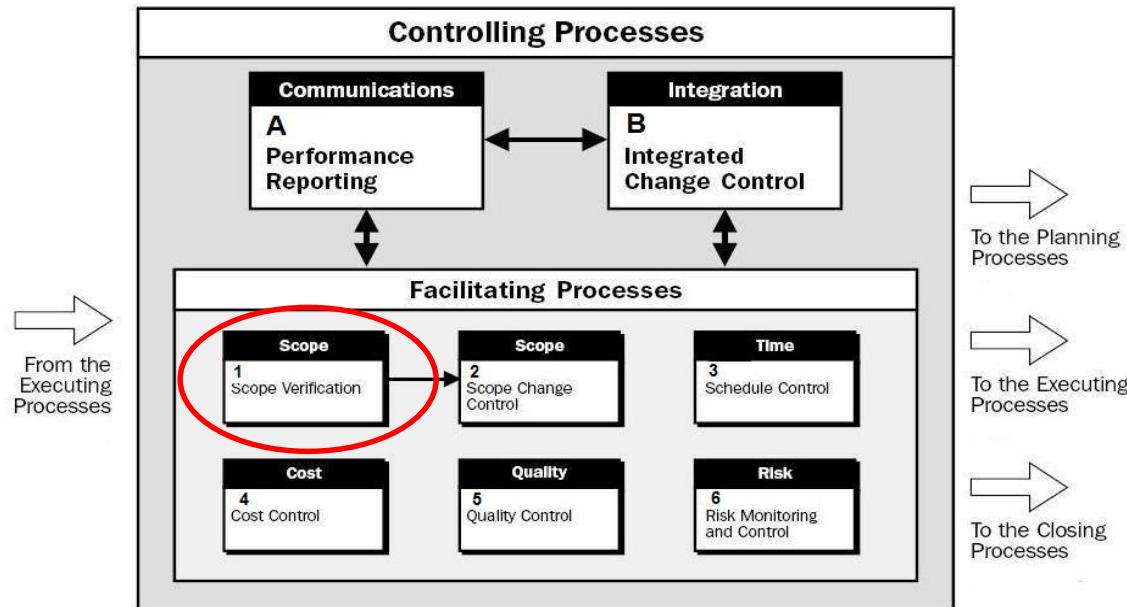
PMC – Example Measures (contd.)

- Budget/Cost performance
 - Compare rate of spending on the project by period (week or month) compared to the planned spending
- Requirements change
 - Track requirements change by period (month usually): total number of requirements, number added in this period, number deleted in this period, and number changed in this period
- EVA – Earned Value Analysis

PMBOK Guide



Scope Verification



- Obtain formal acceptance of the project scope by stakeholders
- Review deliverables – correct and satisfactory (not quality check!!)

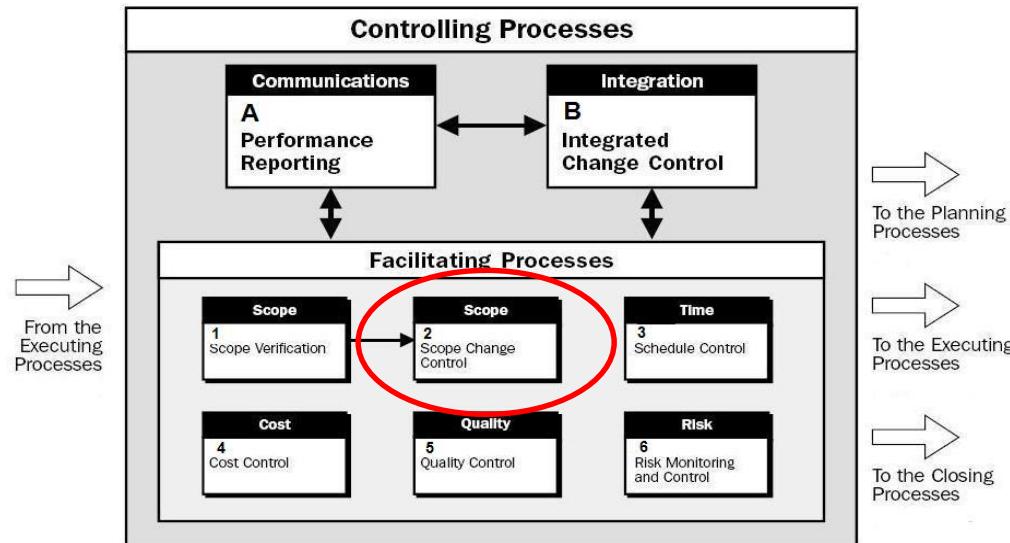
Scope Verification

- Inputs:
 - Work results: which deliverables – fully/partially completed
 - Product documentation: describes project's products
 - WBS: used to verify work of the project; defines scope baseline
 - Scope statement: defines scope in some detail
 - Project plan

Scope Verification

- Outputs:
 - Formal acceptance: must be documented; can be conditional
- Tools & Techniques:
 - Inspection: measure/examine/test results vs. requirements

Scope Change Control



- Concerned with: influencing change factors; determining occurrence of a scope change; managing changes if/when occur
- Must be integrated with other control processes

Scope Change Control

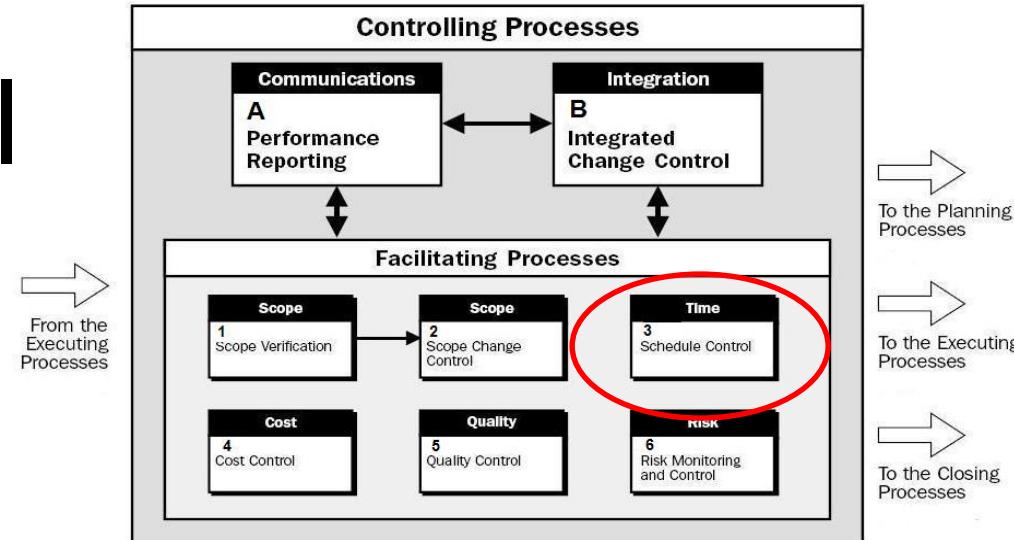
- Inputs:
 - WBS
 - Performance reports: e.g. which interim deliverables have been completed, which not
 - Change requests: oral/written, direct/indirect, external/internal, legally mandated/optional
 - Scope management plan:
 - how scope is managed;
 - how changes are integrated;
 - assessment of expected stability;
 - how changes are identified/classified

Scope Change Control

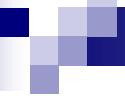
- Outputs:
 - Scope changes
 - Corrective changes
 - Lessons learned
 - Adjusted baseline

- Tools & Techniques:
 - Scope change control
 - Performance measurement
 - Additional planning: e.g. modification to WBS

Schedule Control



- Concerned with: influencing change factors; determining occurrence of a schedule change; managing changes if/when occur [same as scope change control]
- Must be integrated with other control processes



Schedule Control

- Inputs:
 - Project schedule: schedule baseline
 - Performance reports
 - Change requests
 - Schedule management plan

Schedule Control

- Outputs:
 - Schedule updates
 - Corrective action
 - Lessons learned
- Tools & Techniques:
 - Schedule change control system
 - Performance measurement
 - Additional planning
 - Project management software
 - Variance analysis

Schedule Control – metrics

- EVA – Earned Value Analysis
 - Variance analysis – not full picture
 - Planned vs. completed work, to determine if cost, schedule, and work are progressing as planned

Schedule Control – metrics

- Three key values: used to derive further schedule/cost indicators:
 - **BCWS**: budgeted cost of work scheduled; a.k.a. **PV – Planned Value**
 - **ACWP**: actual cost of work performed; a.k.a. **AC – Actual Cost**
 - **BCWP**: budgeted cost of work performed; a.k.a. **EV – Earned Value**

Schedule Control

Schedule		
Cost\Work	Scheduled	Performed
Budgeted	BCWS (PV)	BCWP (EV)
Actual		ACWP (AC)

Schedule Control – metrics

■ Schedule indicators

□ SV: Schedule Variance:

$$\blacksquare \text{SV} = \text{BCWP} - \text{BCWS}$$

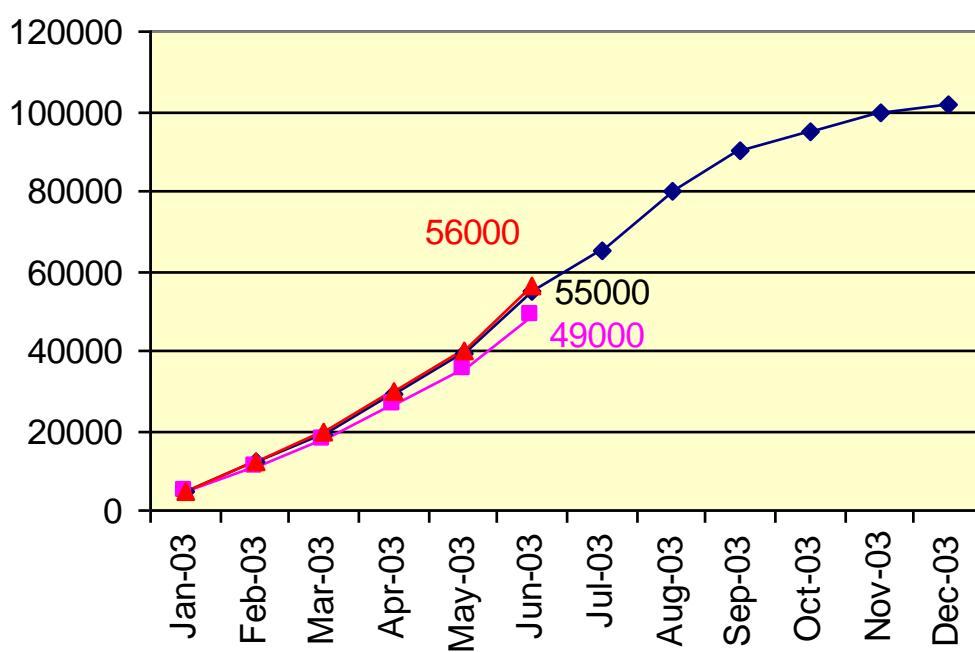
■ < 0 => behind schedule

□ SPI: Schedule Performance Index

$$\blacksquare \text{SPI} = \text{BCWP}/\text{BCWS}$$

■ < 1 => behind schedule

Example



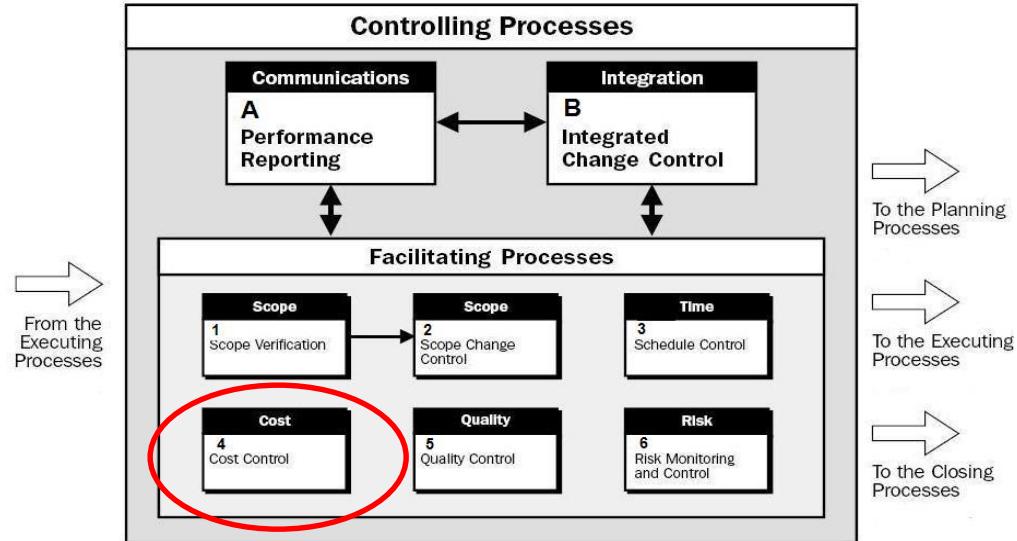
SV = BCWP-BCWS

$$SV = \frac{55,000 - 56,000}{\$49,000} = - \$6,000$$

SPI: BCWP/BCWS

$$49,000 / 55,000 = 0.891$$

Cost Control



- Concerned with: same as before
- Must be integrated with other control processes
- Includes:
 - Monitor cost performance, detect and understand variances from plan
 - Ensure all appropriate changes – recorded accurately in cost baseline

Cost Control

- Includes (contd):
 - Prevent incorrect/inappropriate/unauthorized changes to cost baseline
 - Inform stakeholders
 - Act to bring expected costs within acceptable limits

Cost Control

■ Inputs:

- Cost baseline
- Performance reports
- Change requests
- Cost management plan

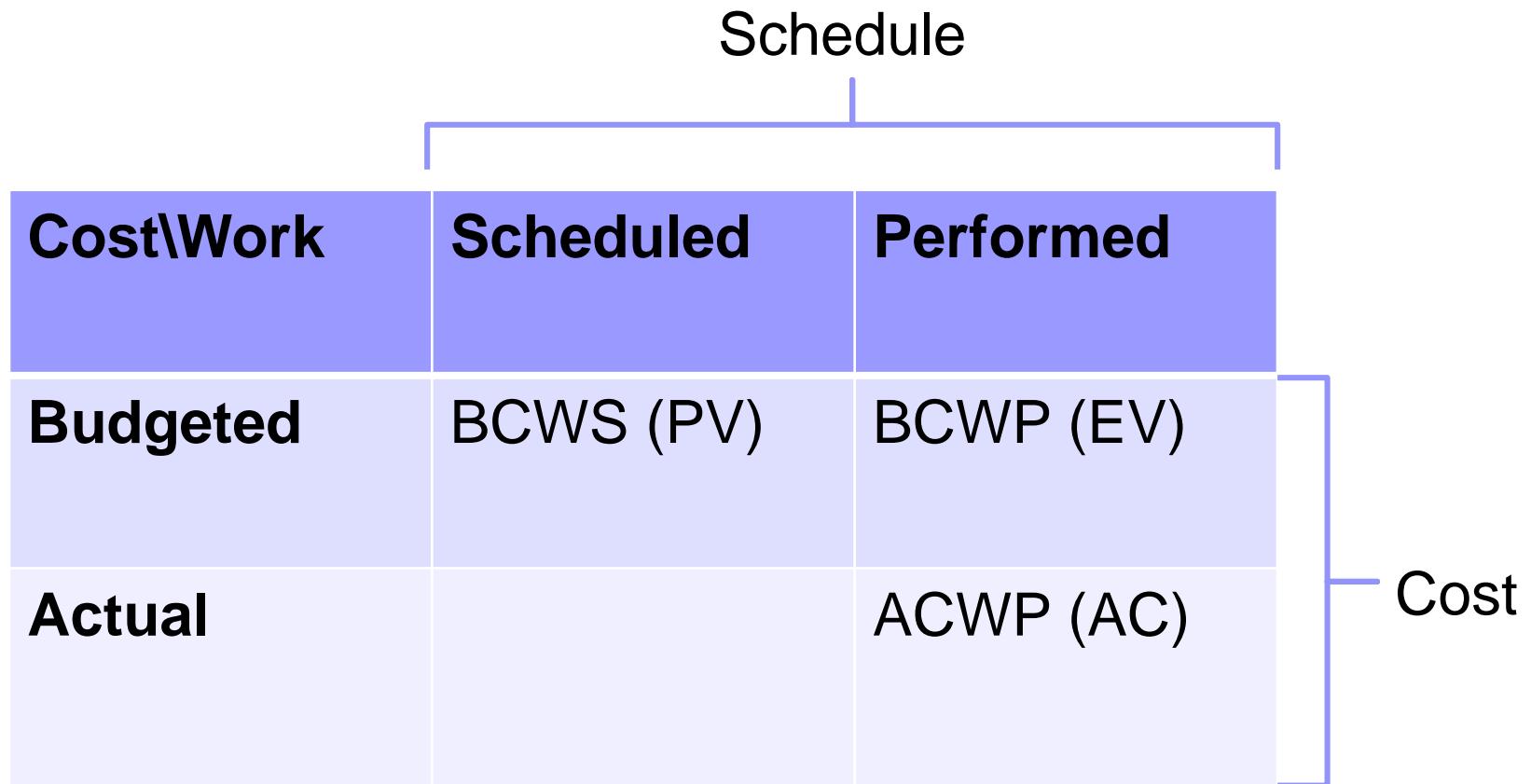
Cost Control

- Outputs
 - Revised cost estimates
 - Budget updates
 - Corrective action
 - Estimates at completion
 - Project closeout
 - Lessons learned

Cost Control

- Tools & Techniques
 - Cost change control system
 - Performance measurement
 - EVA
 - Additional planning
 - Software tools (e.g. MSProject)

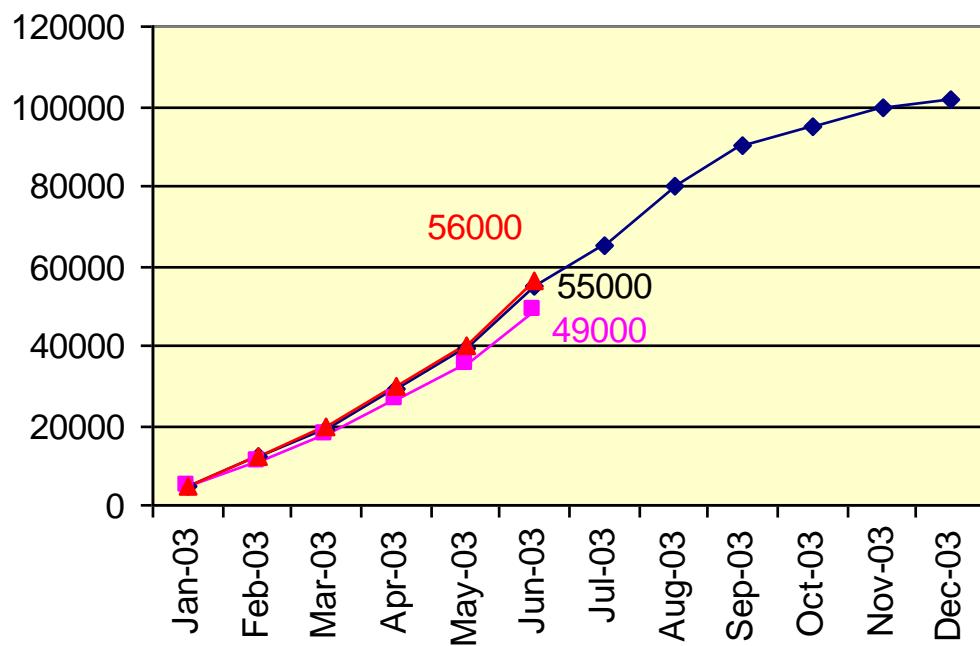
Cost Control



Cost Control – metrics – EVA

- Cost indicators:
 - CV: Cost Variance
 - $CV = BCWP - ACWP$
 - $< 0 \Rightarrow$ over the budget
 - CPI: Cost Performance Index
 - $CPI = BCWP/ACWP$
 - $< 1 \Rightarrow$ over the budget

Example



$$CV = BCWP - ACWP$$

$$\begin{aligned} &\$49,000 - \\ &\underline{56,000} \\ CV &= -\$7,000 \end{aligned}$$

$$CPI: BCWP/ACWP$$

$$49,000/56,000 = 0.875$$

EAC – Estimate At Completion

- Forecast of most likely total project costs
- 3 options depending on the project characteristics
- EAC = actuals to date + new estimate for remaining work
 - EAC = AC + ETC
 - Used when **original assumptions are proven flawed/no longer relevant** to a change in conditions

EAC – Estimate At Completion

- EAC = actuals to date + remaining budget
 - $EAC = AC + (BAC - BCWP)$;
 - Used when **current variances are atypical; not expected to appear in future**

EAC – Estimate At Completion

- EAC = actual to date + remaining budget modified by a performance factor (often CPI)
 - $EAC = AC + (BAC - BCWP)/CPI$
 - Used when **current variances are typical of future variances**
 - Once a project is 20% complete, the CPI does not vary from its current value by more than 10%.

Earned Value Chart

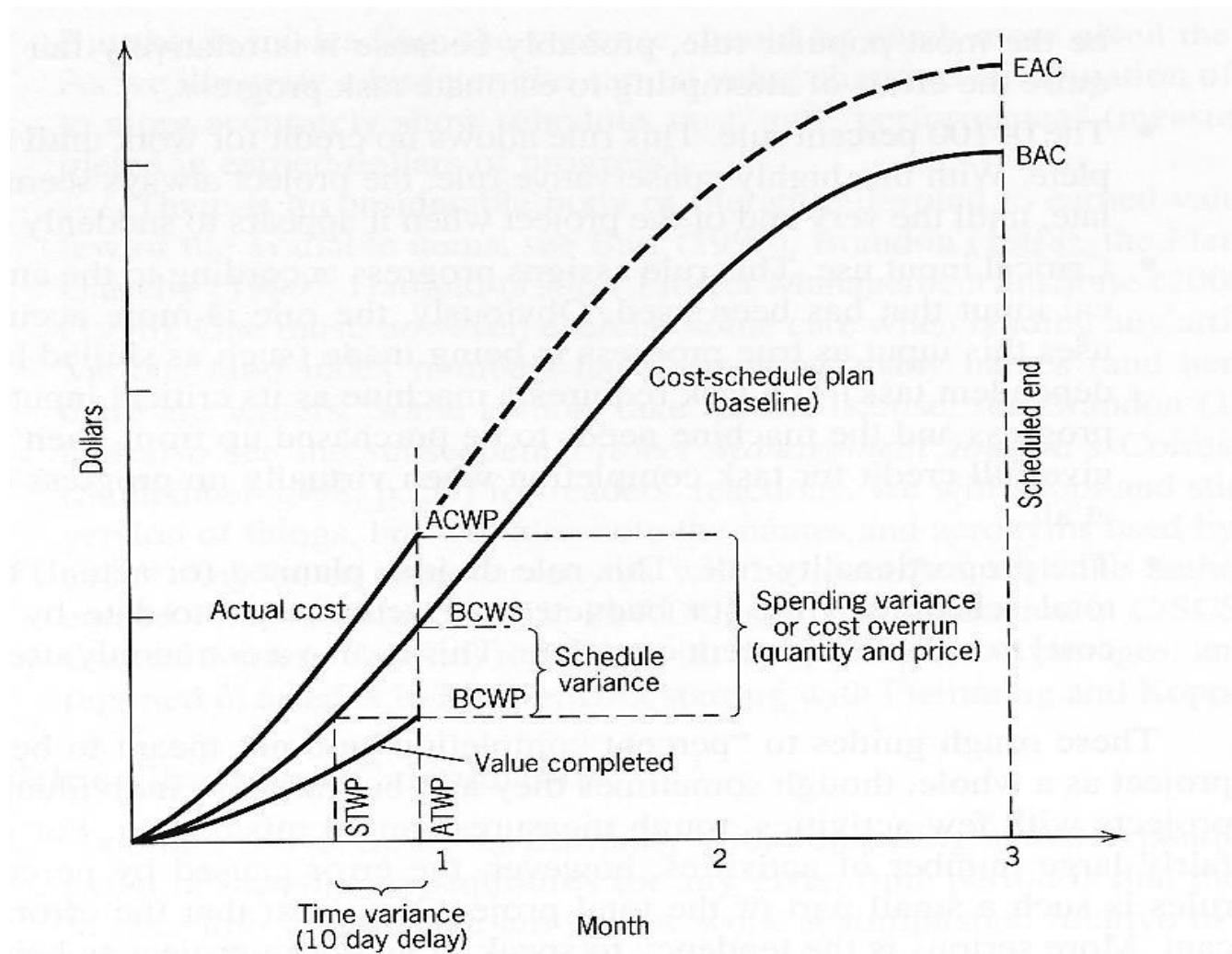


Figure 10-7 Earned value chart.

Example

Assume that operations on a Work Package cost \$1,500 to complete. They were originally scheduled to finish today. At this point, we actually spent \$1,350. And we estimate that we have completed two thirds (2/3) of the work. What are the cost and schedule variances?

$$\begin{aligned} CV &= BCWP - ACWP = 1500 \left(\frac{2}{3}\right) - 1350 \\ &= -350 \end{aligned}$$

$$\begin{aligned} SV &= BCWP - BCWS = 1500 \left(\frac{2}{3}\right) - 1500 \\ &= -500 \end{aligned}$$

Example continued

$CPI = BCWP/ACWP = 1500(2/3)/1350 =$
0.74

$SPI = BCWP/BCWS = 1500(2/3)/1500 =$
0.67

Spending higher than budget, and given what we have spent, we are not as far along as we should be (have **not completed as much work as we should have**)

Relations between indicators

- Possible to have one of indicators to be favorable while the other unfavorable
- Might be ahead of schedule and behind costs
- Six possibilities (see figure next slide)

6 Possibilities Earned Value Analysis

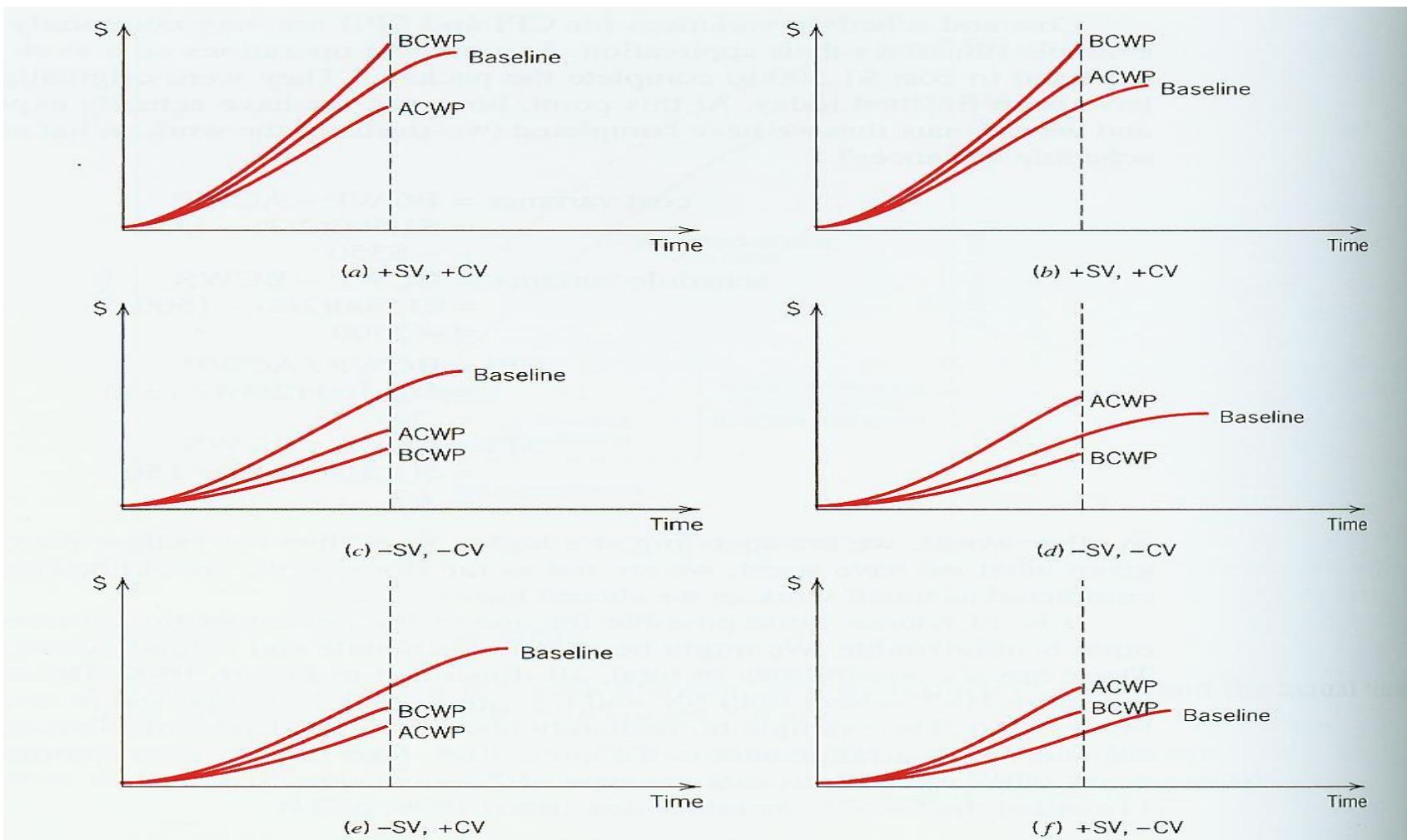


Figure 10-8 Six possible arrangements of ACWP, BCWP, and Baseline resulting in four combinations of positive and negative schedule variance (SV) and cost variance (CV). (Figure 10-7 is arrangement d.)

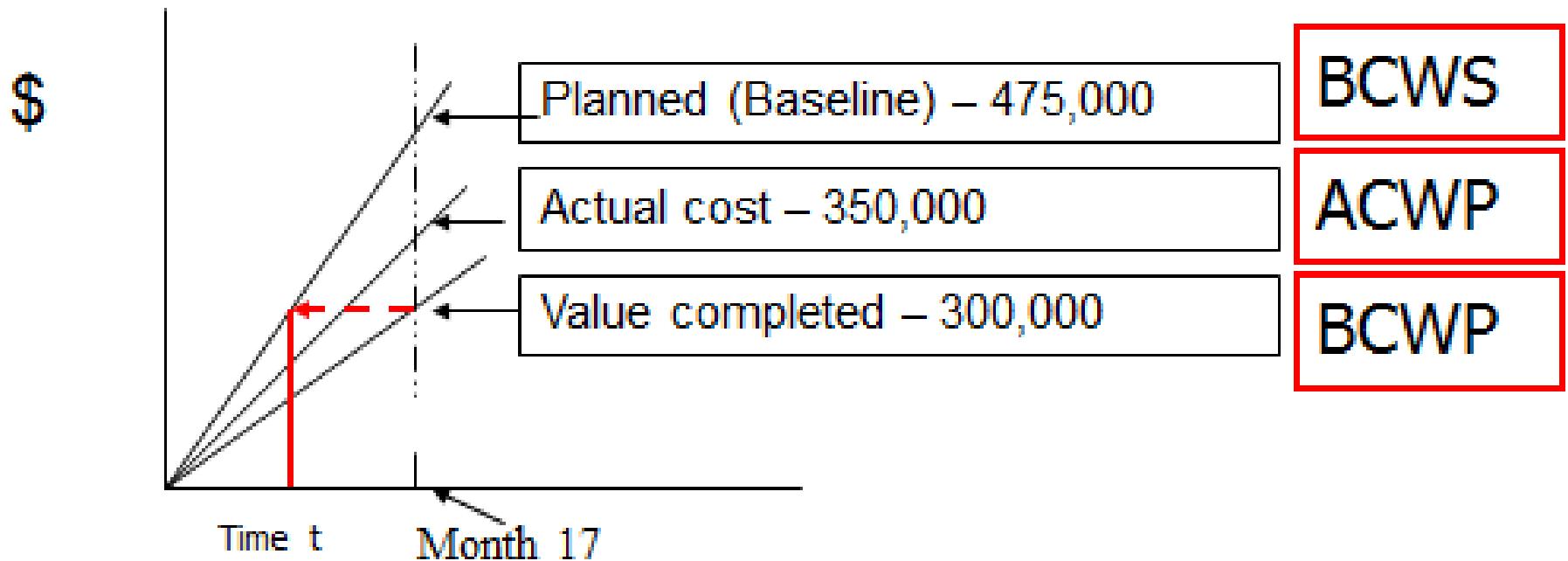
Exercise

- A project to develop a software system has in month 17:
 - an actual cost (AC) of \$350,000,
 - a planned value (PV) of \$475,000,
 - and an earned value (EV) of \$300,000.

Find the cost and schedule variances and the two indexes.

What can you say about the project?

Solution



$$BCWS = 475,000$$

$$BCWP = 300,000$$

$$ACWP = 350,000$$

$$CV = BCWP - ACWP = -50,000$$

$$SV = BCWP - BCWS = -175,000$$

$$CPI = BCWP/ACWP = 0.86$$

$$SPI = BCWP/BCWS = 0.63$$

Critical Ratio

- Sometimes, especially large projects, it may be worthwhile calculating a set of critical ratios for all project activities
- The critical ratio is
$$\frac{\text{actual progress}}{\text{scheduled progress}} \times \frac{\text{budgeted cost}}{\text{actual cost}}$$
- If ratio is 1 everything is probably on target
- The further away from 1 the ratio is, the more we may need to investigate

Critical ratio example

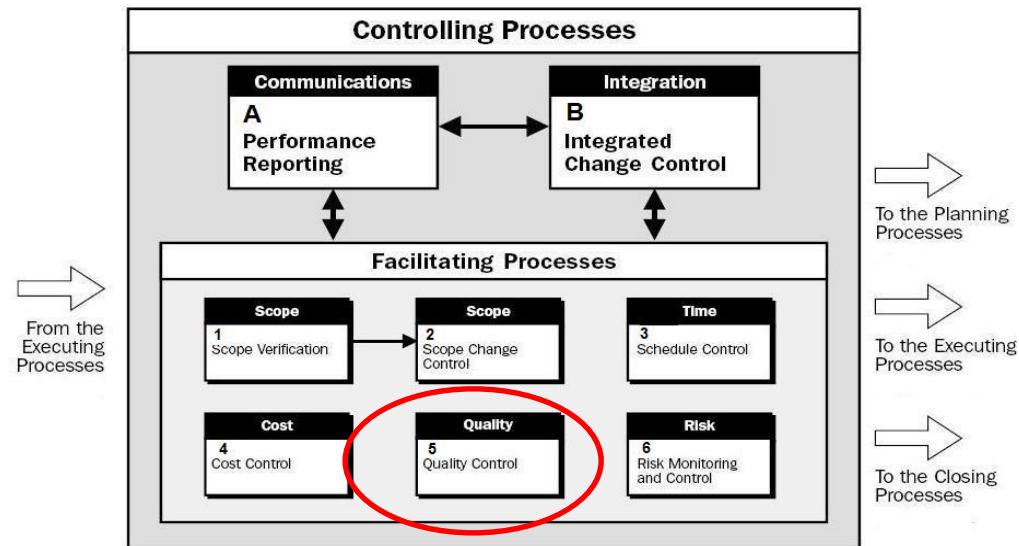
Which activities are probably on target and which need to be investigated?

Activity	Actual progress	Scheduled Progress	Budgeted Cost	Actual cost	Critical ratio (CR)
A	4 days	4 days	60	40	$4/4 * 6/4$
B	3 days	2 days	50	50	$3/2 * 5/5$
C	2 days	3 days	30	20	$2/3 * 3/2$
D	1 day	1 day	20	30	$1/1 * 2/3$
E	2 days	4 days	25	25	$2/4 * 1/1$

Critical ratio example

- Can be on schedule and below budget
(Act A) Why so good? Cutting corners?
- Can be behind schedule but below budget
(Act C)
- Can be on budget but physical progress lagging (Act E)
- Can be on schedule but cost running higher than budget (Act D)
- On budget ahead of schedule (Act B)

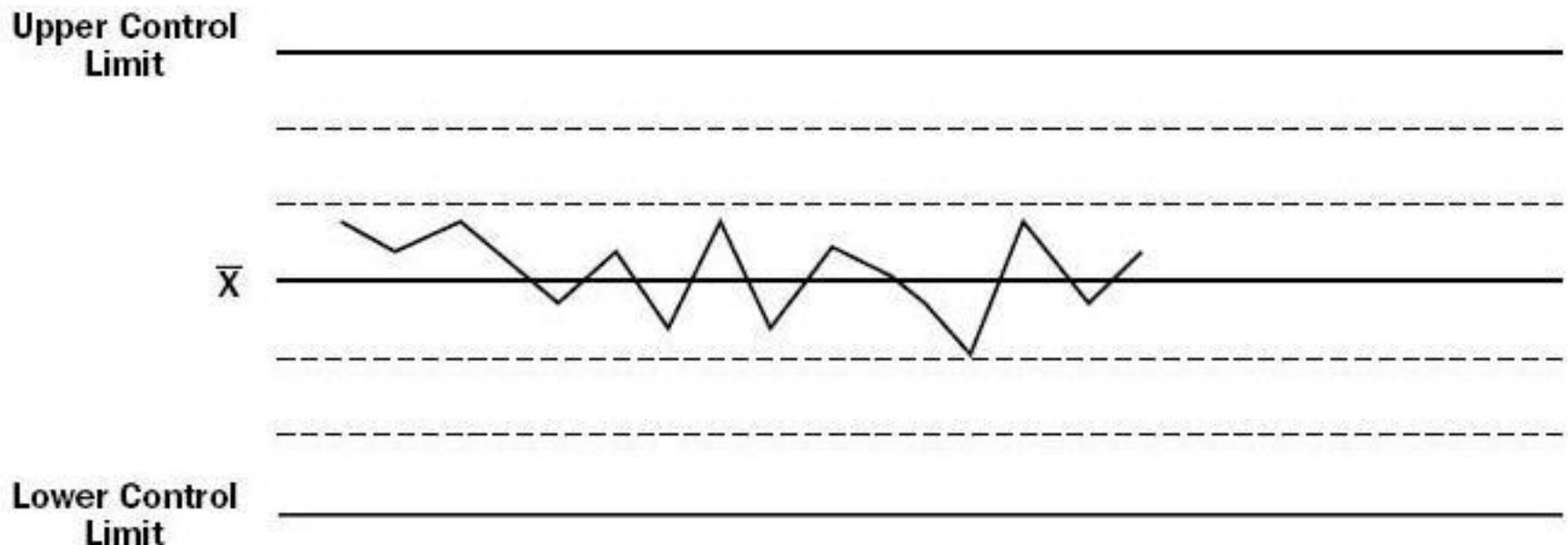
Quality Control



- Monitor and evaluate specific results against relevant quality standards; eliminate causes of unsatisfactory results.
- Often performed by a Quality Control Department

Quality Control – metrics

- **Control charts** = graphic display of results, over time, of a process.
- Used to monitor any type of output variable (cost/schedule variances, scope changes volume/frequency, errors in docs)



The x axis of all control charts consists of sample numbers (usually the time of the sample).

Control chart of project schedule performance

Quality Control – metrics

- Pareto diagrams = histogram, ordered by frequency of occurrence, that shows how many results were generated by type of cause
 - Rank – used to guide corrective action
 - Pareto's law: 80% of problems are due to 20% of causes

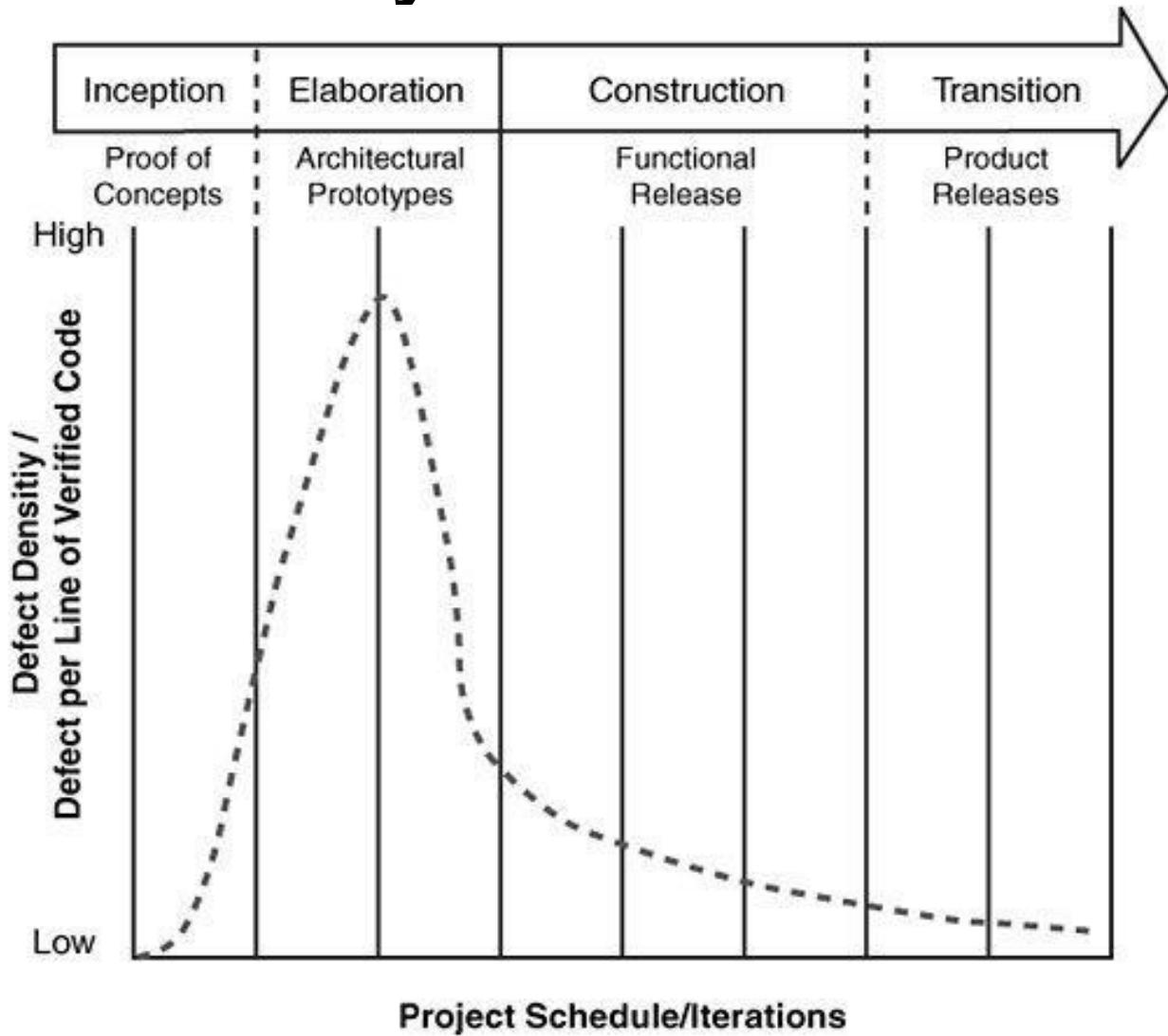
Quality Control – metrics

- Flowcharting = show relations between system elements.
 - Cause-and-effect: how various factors may be linked to potential problems/effects
 - System/process flowcharts

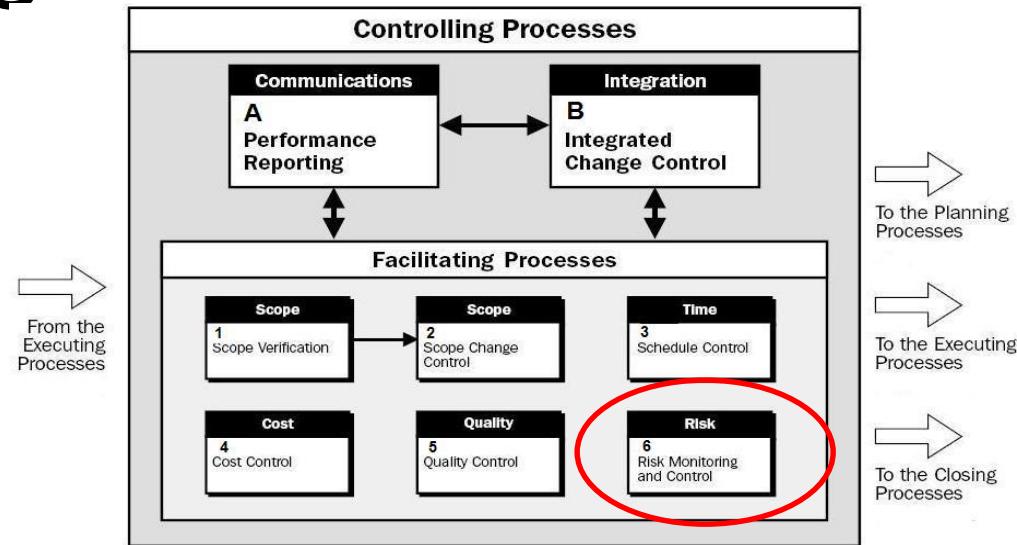
Quality Control – metrics

- Trend Analysis = use techniques to forecast future based on past (correlations)
 - Technical performance: no. of errors identified; no. corrected
 - Cost/Schedule performance: no. of activities/period completed with significant variances

Quality Control – normal trends



Risk Monitoring & Control



- Keep track of identified risks; monitor residual risks; identify new risks; ensure execution of risk plans & evaluate their effectiveness
- Risks change as project matures

Risk Monitoring & Control

■ Purpose – check if:

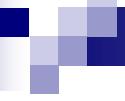
- Risk responses have been implemented as planned
- Risk responses are effective, or new must be implemented
- Project assumptions – still valid
- Risk exposure has changed (trend analysis)

Risk Monitoring & Control

- Purpose – check if:
 - Risk trigger has occurred
 - Proper policies and procedures are followed
 - Risks not previously identified have occurred

Risk Monitoring & Control

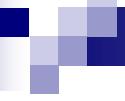
- Inputs:
 - Risk management plan
 - Risk response plan
 - Project communication: issues logs, action-item lists, jeopardy warnings, escalation notices.
 - Additional risk identification and analysis
 - Scope changes



Risk Monitoring & Control

■ Tools & Techniques

- Project risk response audits
- Periodic project risk reviews
- Earned Value Analysis
- Technical performance measurement
- Additional risk response planning

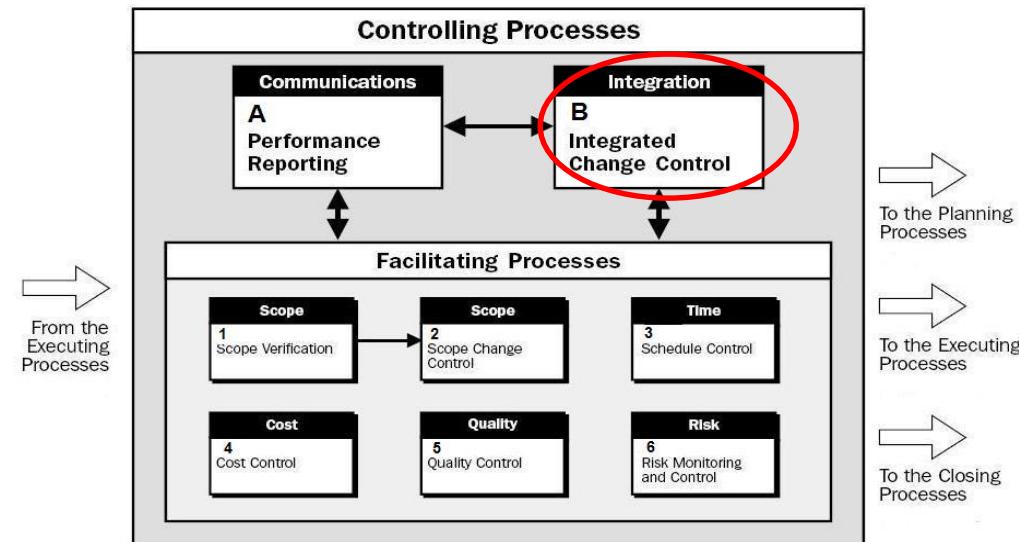


Risk Monitoring & Control

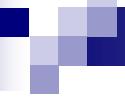
■ Outputs:

- Workaround plans
- Corrective action
- Project change requests
- Updates to the risk response plan
- Risk database
- Updates to risk identification checklists

Integrated Change Control



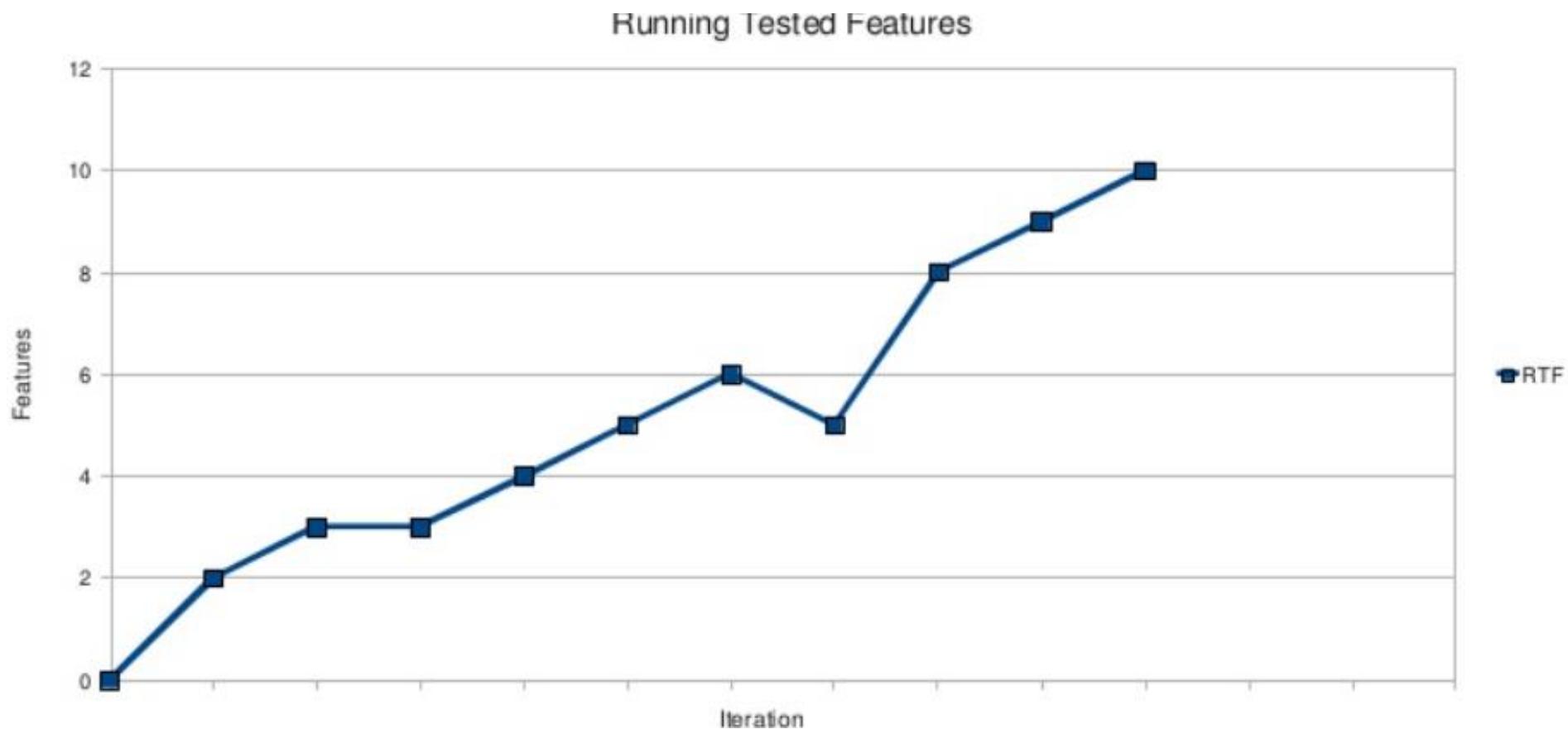
- Concerned with: influencing change factors; determining occurrence of a change; managing changes if/when occur
- Original project scope and performance baseline – maintained by continuously managing changes



Agile metrics are about...

- Measuring outcome, not activity
- Principles that guide the choice of metrics
 - “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software”
 - “Working software is the primary measure of progress”

Running tested features



Running tested features

- Principle
 - Working software is the primary measure of progress
- Informational
 - Direct measure of delivered results
- Diagnostic
 - If flat or declining over time indicates a problem
- Motivational
 - Team members naturally want to see RTF increase

Earned Business Value

■ Principle

- Highest priority is to satisfy customer through early and continuous delivery of valuable software

■ Informational

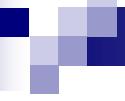
- Direct measure of customer-defined value delivered

■ Diagnostic

- Trend should be a steep curve

■ Motivational

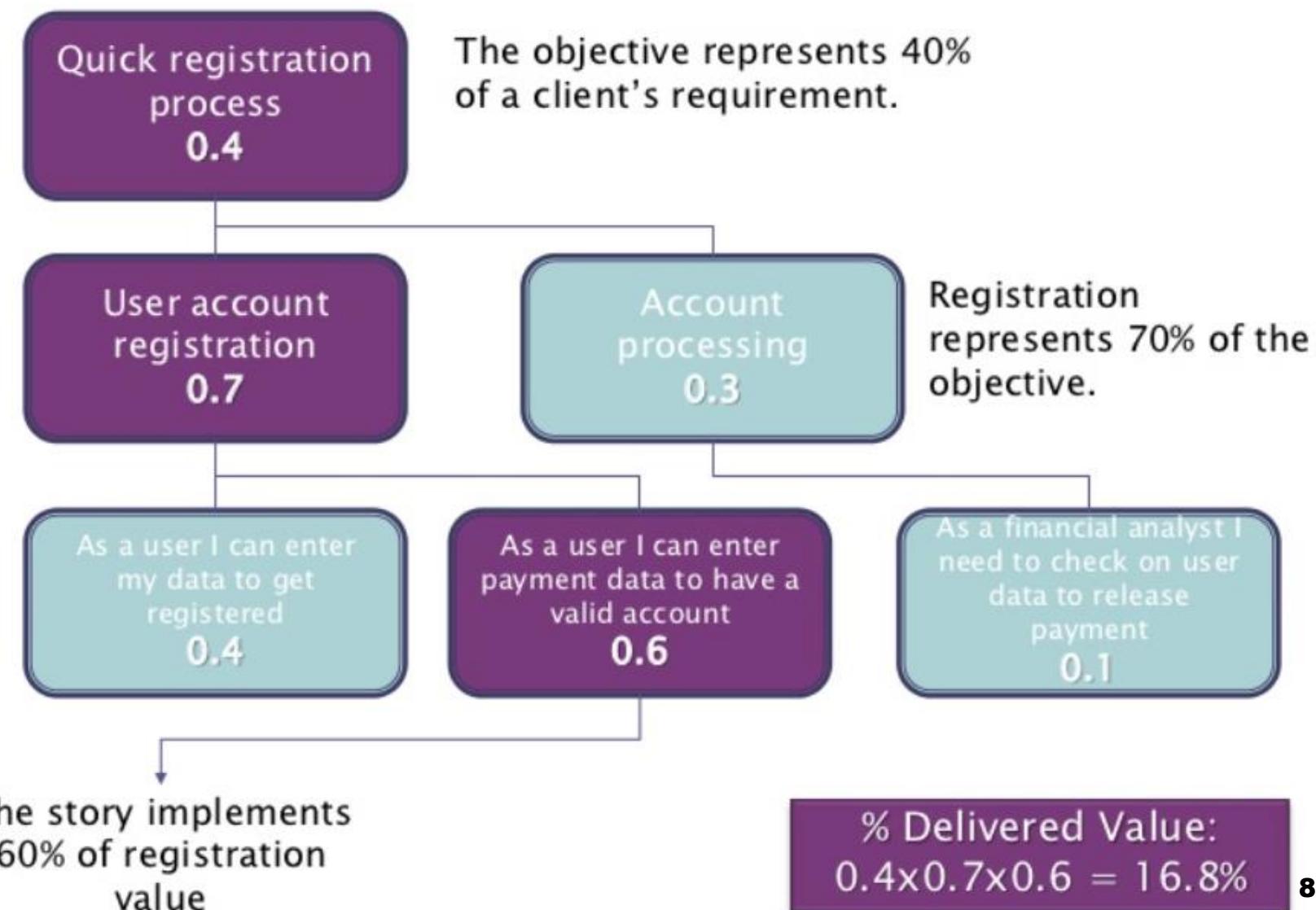
- Team members like to deliver value



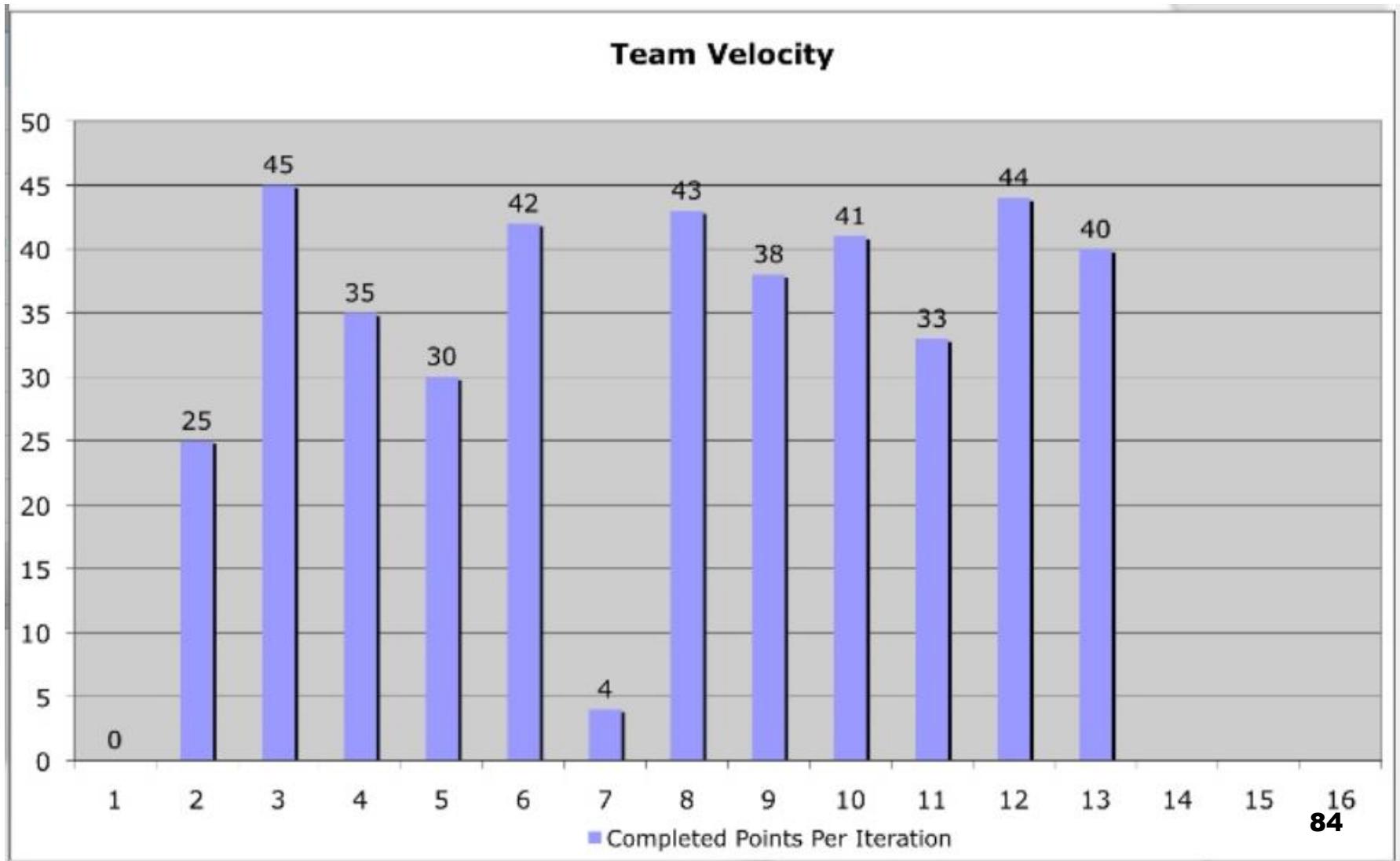
Business value forms

- Revenue
- Cost savings
- Market share
- Customer relations
- Reputation
- Other ☺

Calculating Business value



Measuring velocity



Velocity

■ Principle

- Highest priority is to satisfy customer through early and continuous delivery of valuable software

■ Informational

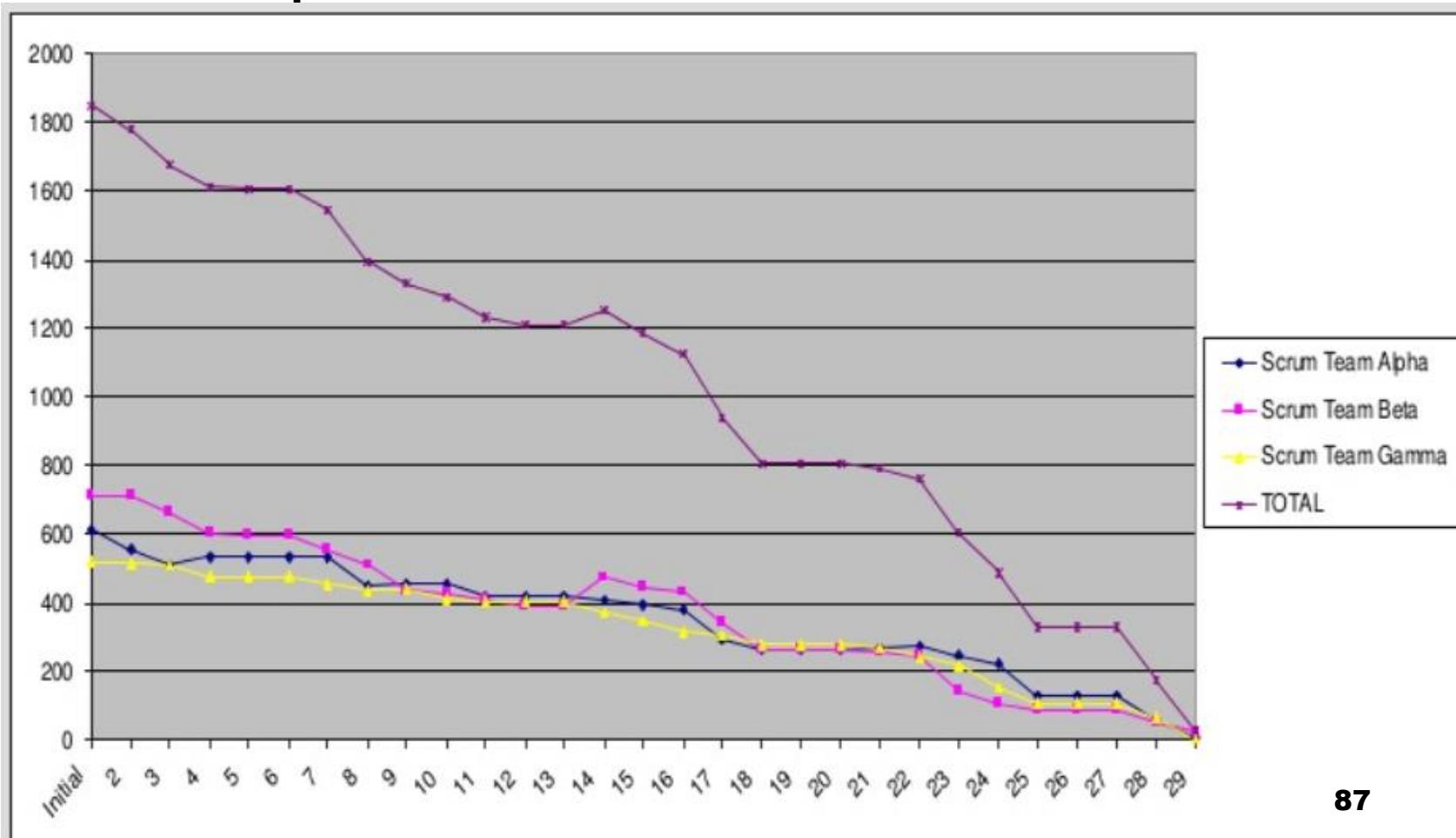
- Empirical observation of the team's capacity for work
- Helps projecting the likely completion date of a given amount of scope
- Helps estimating the amount of scope that can be delivered by a given date

Velocity

- Diagnostic
 - Patterns in trends indicate various problems
 - Provides a baseline for improvement effort
- Motivational
 - Team members take pride in achieving high velocity and keeping it stable

Burndown charts

- Tell us in a visual way how much work has been completed.



Earned Value Method in Agile

- Initial baseline calculated based on 5 data points:
 - The number of planned sprints in the release backlog
 - The length of each sprint in calendar days
 - The number of story points planned for the release backlog
 - The budget planned for the release backlog
 - The start date of the project

EVM

Item	Definition
Budget at Complete (BAC)	The planned amount you expect to spend
Actual Cost (AC)	The actual cost to complete the work
Planned release story points for the release (PRSP)	Story points are defined at the product backlog level
Expected Percent Complete (EPC)	Current sprint(n) / Total planned sprints
Actual Percent Complete (APC)	Story points completed / Total planned story points
Planned Value (PV)	$PV = BAC * EPC$
Earned Value (EV)	$EV = BAC * APC$
Cost Performance Index (CPI)	$CPI = EV / AC$
Schedule Performance Index (CSPI)	$CSPI = EV / PV$

Conclusions

- The progress towards project goals and the quality of software products must be measurable throughout the software development lifecycle
- Metrics values provide an important perspective for managing the process. Metrics trends provide another.

Conclusions

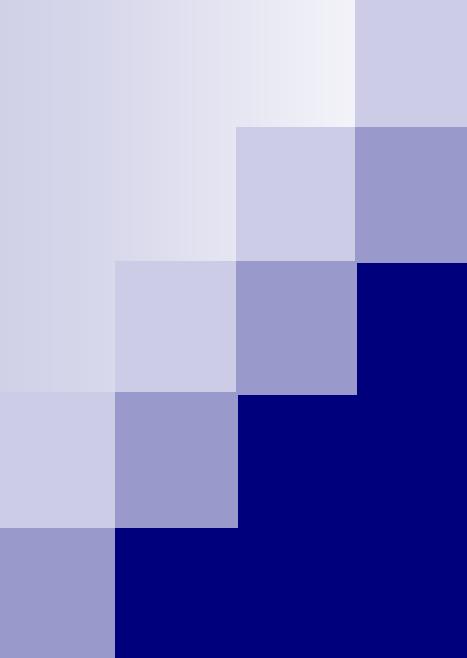
- The most useful metrics are extracted directly from the evolving artefacts
- Objective analysis and automated data collection are crucial for any metrics program.
- Subjective assessments and manual data collection are likely to fail.

Quiz Time

- Let's switch to Moodle

Discussion for next time

- Give a Earned Value Analysis example by showing the current status against the baseline (estimation).
- Discuss the status of the project (ahead/behind schedule, over/under the budget).



People Management

Lecture 10

...recap::Monitoring and Control Definition

- The goal of Project Monitoring and Control (PMC) is to provide an **understanding of the project's progress** so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.

..recap: Monitoring and Control Goals

- Provide project manager and development team with the following:
 - An accurate **assessment of the progress** to date
 - Insight into the **quality** of the evolving software product
 - A **basis for estimating** cost and budgeting with increased accuracy over time

...recap::Monitoring and Control Deliverables

- Written status reports
- Updates to lists of action items, risks, problems, and issues
- Updates to plan/schedule, to reflect actual progress
- Comparison of actual vs. budgeted cost, cost/benefit analysis (EVA)
- Audit/review reports of the activities and work products under development

...recap::Monitoring and Control– Example Measures

- Milestone attainment
- Effort spent
- Budget/Cost performance
- Requirements change
- EVA – Earned Value Analysis

...recap::Monitoring and Control

- Control Metrics

- Measurement done at WBS unit level:
 - CAP – Intersection between WBS and OBS
 - CAP – Control Account Plan = work package with extra features:
 - Assignment of responsibility (organization/individual)
 - Division (if needed) in lower-level work packages
 - Metrics for measuring performance: milestones, %complete...

...recap::Monitoring and Control

- Control Metrics

EVA – Earned Value Analysis

Schedule		
Cost\Work	Scheduled	Performed
Budgeted	BCWS	BCWP
Actual		ACWP

Cost

...recap::Monitoring and Control

- Control Metrics EVA

- Three key values: used to derive further schedule/cost indicators:
 - BCWS: budgeted cost of work scheduled; a.k.a. PV – Planned Value
 - ACWP: actual cost of work performed; a.k.a. AC – Actual Cost
 - BCWP: budgeted cost of work performed; a.k.a. EV – Earned Value

...recap::Monitoring and Control Schedule Control Metrics – EVA

- Schedule indicators
 - SV: Schedule Variance:
 - $SV = BCWP - BCWS$
 - $< 0 \Rightarrow$ behind schedule
 - SPI: Schedule Performance Index
 - $SPI = BCWP/BCWS$
 - $< 1 \Rightarrow$ behind schedule

...recap::Monitoring and Control

- Cost Control Metrics – EVA

- Cost indicators:

- CV: Cost Variance

- $CV = BCWP - ACWP$
 - $< 0 \Rightarrow$ over the budget

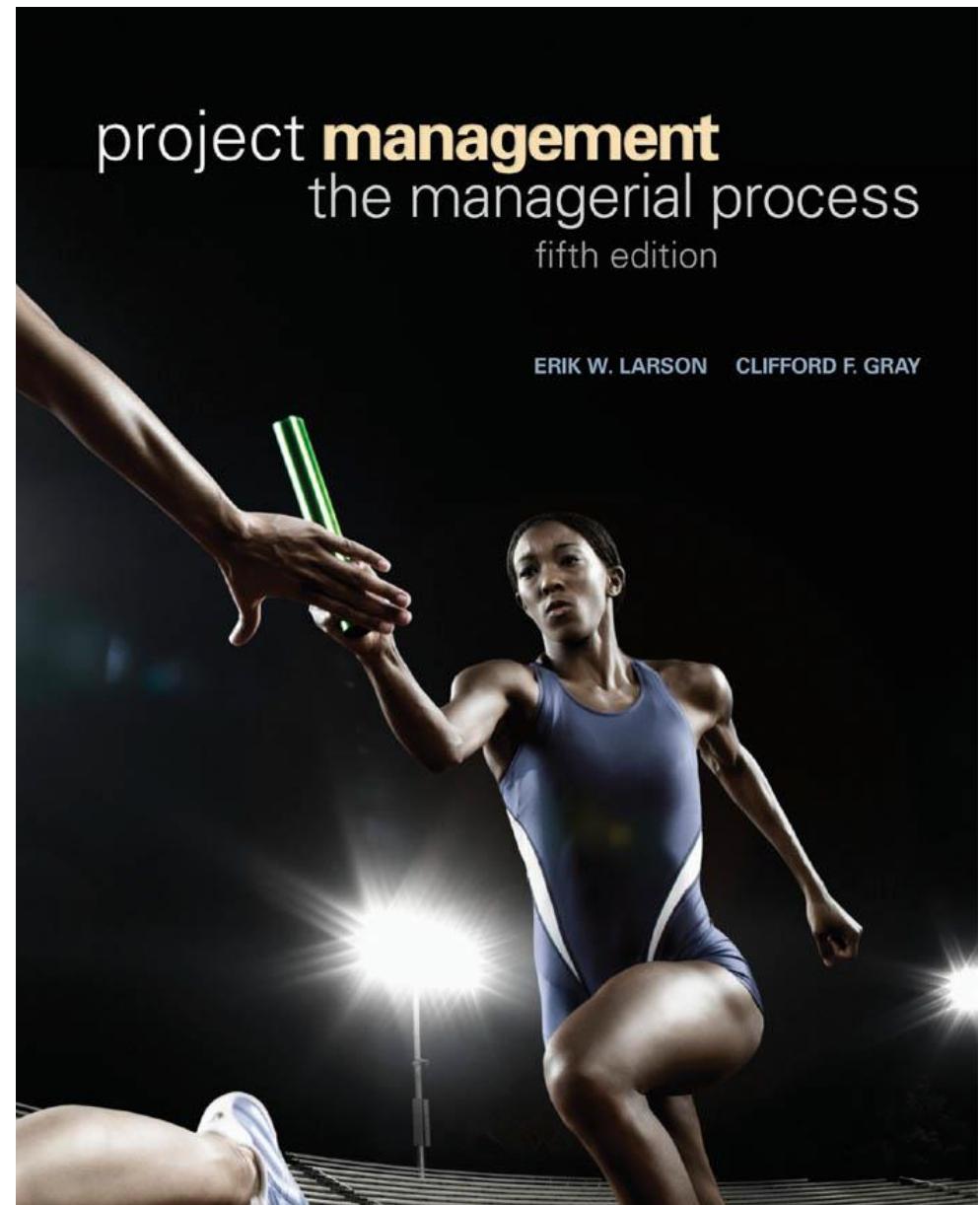
- CPI: Cost Performance Index

- $CPI = BCWP/ACWP$
 - $< 1 \Rightarrow$ over the budget

Today's topic

- People Management
 - Leadership
 - Manage teams

References



Managing versus Leading a Project

- **Managing—coping with complexity**
 - Formulate plans and objectives
 - Monitor results
 - Take corrective action
 - Expedite activities
 - Solve technical problems
 - Serve as peacemaker
 - Make tradeoffs among time, costs, and project scope
- **Leading—coping with change**
 - Recognize the need to change to keep the project on track
 - Initiate change
 - Provide direction and motivation
 - Innovate and adapt as necessary
 - Integrate assigned resources

Managing Project Stakeholders

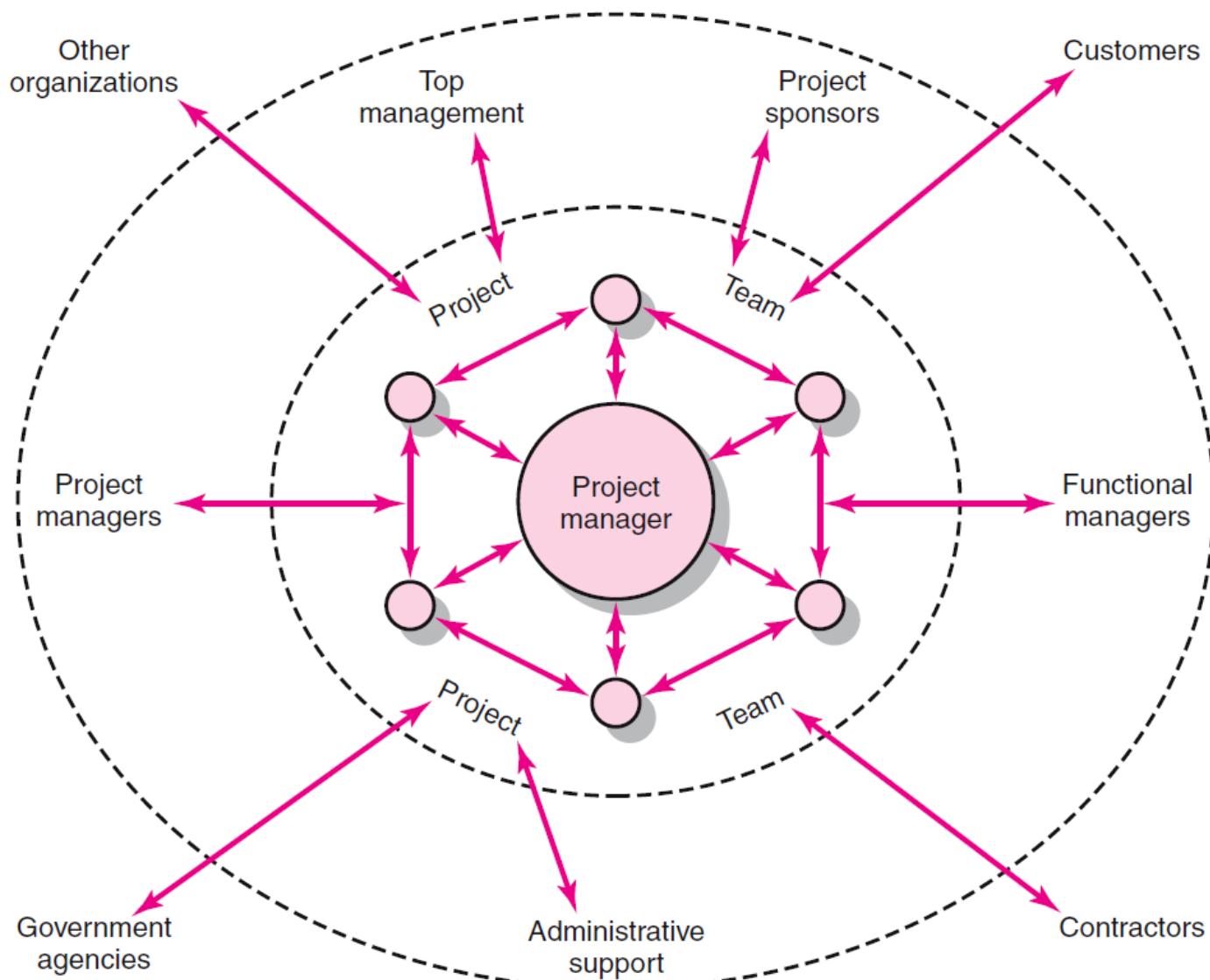
■ Project Management Maxims:

- You can't do it all and get it all done
 - Projects usually involve a vast web of relationships.
- Hands-on work is not the same as leading.
 - More pressure and more involvement can reduce your effectiveness as a leader.

Project Management Maxims

- What's important to you likely isn't as important to someone else
 - Different groups have different stakes (responsibilities, agendas, and priorities) in the outcome of a project.
- Remember: project management is a tough, exciting, and rewarding—endeavor to persevere.

Network of Stakeholders



Influence as Exchange

- The Law of Reciprocity
 - One good deed deserves another, and likewise, one bad deed deserves another.
- Quid pro Quo
 - Mutual exchanges of resources and services (“back-scratching”) build relationships.
- Influence “Currencies” (Cohen and Bradford)
 - Cooperative relationships are built on the exchange of organizational “currencies” (favors).

Commonly Traded Organizational Currencies

Task-related currencies

Resources	Lending or giving money, budget increases, personnel, etc.
Assistance	Helping with existing projects or undertaking unwanted tasks.
Cooperation	Giving task support, providing quicker response time, or aiding implementation.
Information	Providing organizational as well as technical knowledge.

Position-related currencies

Advancement	Giving a task or assignment that can result in promotion.
Recognition	Acknowledging effort, accomplishments, or abilities.
Visibility	Providing a chance to be known by higher-ups or significant others in the organization.
Network/contacts	Providing opportunities for linking with others.

Source: Adapted from A. R. Cohen and David L. Bradford, *Influence without Authority* (New York: John Wiley & Sons, 1990). Reprinted by permission of John Wiley & Sons, Inc.

Organizational Currencies

Inspiration-related currencies

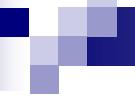
Vision	Being involved in a task that has larger significance for the unit, organization, customer, or society.
Excellence	Having a chance to do important things really well.
Ethical correctness	Doing what is “right” by a higher standard than efficiency.

Relationship-related currencies

Acceptance	Providing closeness and friendship.
Personal support	Giving personal and emotional backing.
Understanding	Listening to others’ concerns and issues.

Personal-related currencies

Challenge/learning	Sharing tasks that increase skills and abilities.
Ownership/involvement	Letting others have ownership and influence.
Gratitude	Expressing appreciation.



Social Network Building

■ Mapping Dependencies

- Project team perspective:

- Whose cooperation will we need?
 - Whose agreement or approval will we need?
 - Whose opposition would keep us from accomplishing the project?

- Stakeholders' perspective:

- What differences exist between the team and those on whom the team will depend?
 - How do the stakeholders view the project?
 - What is the status of our relationships with the stakeholders?
 - What sources of influence does the team have relative to the stakeholders?

Management by Wandering Around

- Managers spend the majority of their time in face-to-face interactions with employees building cooperative relationships.
- Characteristics of Effective Project Managers
 - **Initiate** contact with key players.
 - **Anticipate** potential problems.
 - **Provide** encouragement.
 - **Reinforce** the objectives and vision of the project.
 - **Intervene** to resolve conflicts and prevent stalemates.²¹

Manage Expectations

- The way you present information can either clarify or muddy expectations.
 - For example, you estimate that a task will take 317 hours – will you promise 317 or [300 – 325]?
- Recognize that it is only human nature to interpret a situation in one's best interest.
 - For example, if you tell someone it will be done by January....
- Seize every opportunity to realign expectations with reality.

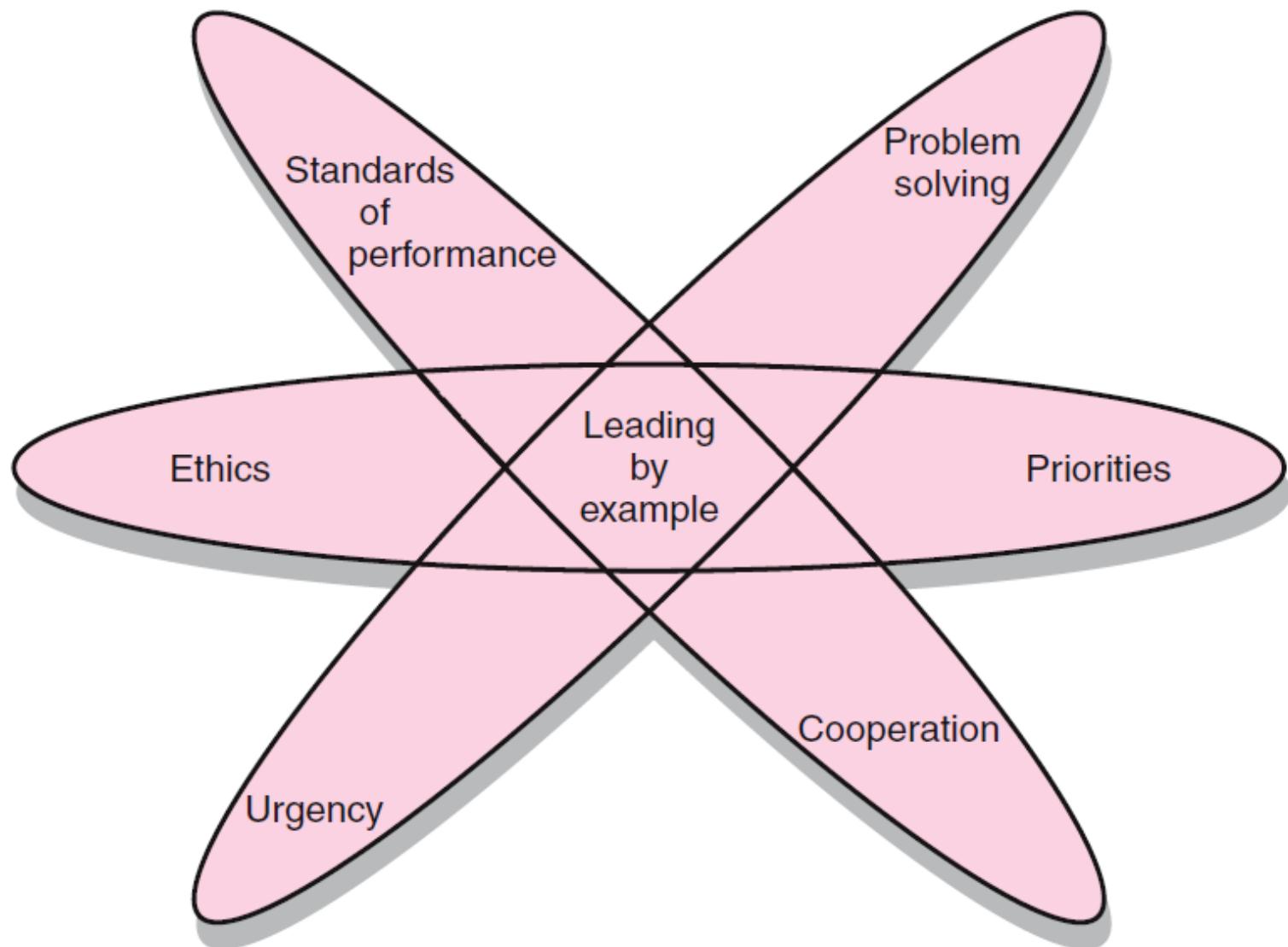
Manage Expectations

- Do not ask for stakeholder suggestions for improvement if you do not intend to do something with their input.
- State the obvious. What is obvious to you may be obscure to others.
- Don't avoid delivering bad news.
 - Communicate openly and in person.
 - Expect some anger and frustration. Do not get defensive in return.
 - Be prepared to explain the impact of the problems.

Managing Upward Relations

- Project Success = Top Management Support
 - Appropriate budgets
 - Responsiveness to unexpected needs
 - A clear signal to the organization of the importance of cooperation
- Motivating the Project Team
 - Influence top management in favor of the team:
 - Rescind unreasonable demands
 - Provide additional resources
 - Recognize the accomplishments of team members

Leading by Example



Ethics and Project Management

■ Ethical Dilemmas

□ Situations where it is difficult to determine whether conduct is right or wrong:

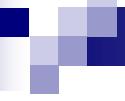
- Padding of cost and time estimations
- Exaggerating pay-offs of project proposals
- Falsely assuring customers that everything is on track
- Being pressured to alter status reports
- Falsifying cost accounts
- Compromising safety standards to accelerate progress
- Approving shoddy work

□ Code of conduct

- Professional standards and personal integrity

Trust

- Ethic => Trust
- Character (Does he want the right thing?)
 - ⇒ Consistency (predictable)
 - ⇒ Openness
 - ⇒ Sense of purpose



Trust

- Competence (Does he know the right thing to do?)
 - Tasks
 - Interpersonal
 - Organizational

Contradictions of Project Management

- Innovate and maintain stability.
- See the big picture while getting your hands dirty.
- Encourage individuals but stress the team.
- Hands-off/Hands-on.
- Flexible but firm.
- Team versus organizational loyalties.

Qualities of an Effective Project Manager

1. Systems thinker
2. Personal integrity
3. Proactive
4. High emotional intelligence (EQ).
5. General business perspective
6. Effective time management
7. Skillful politician
8. Optimist

EQ

- Self-awareness
- Self-regulation
- Self-motivation
- Empathy
- Social skills

Suggestions for Project Managers

- Build relationships before you need them.
- Trust is sustained through frequent face-to-face contact.
- Realize that “what goes around comes around.”



High-Performing Teams

■ Synergy

- $1 + 1 + 1 = 10$ (positive synergy)
- $1 + 1 + 1 = 2$ (negative synergy)

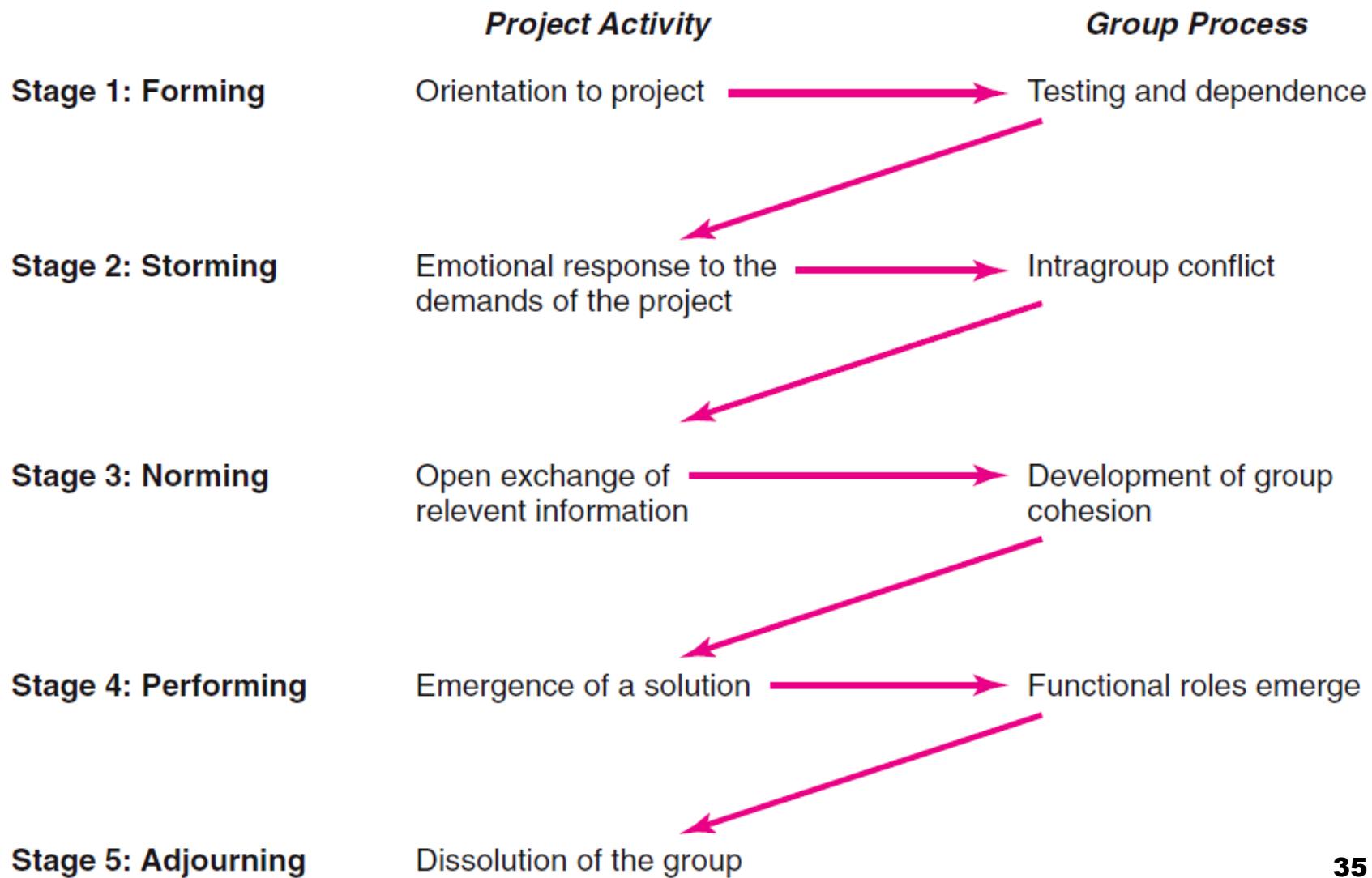
“The difference in productivity between an average team and a turned-on, high-performing team is not 10 percent, 20 percent, or 30 percent, but 100 percent, 200 percent, even 500 percent!”

—Tom Peters, management consultant and writer

Characteristics of High-performing Teams

1. Share a sense of common purpose
2. Make effective use of individual talents and expertise
3. Have balanced and shared roles
4. Maintain a problem solving focus
5. Accept differences of opinion and expression
6. Encourage risk taking and creativity
7. Sets high personal performance standards
8. Identify with the team

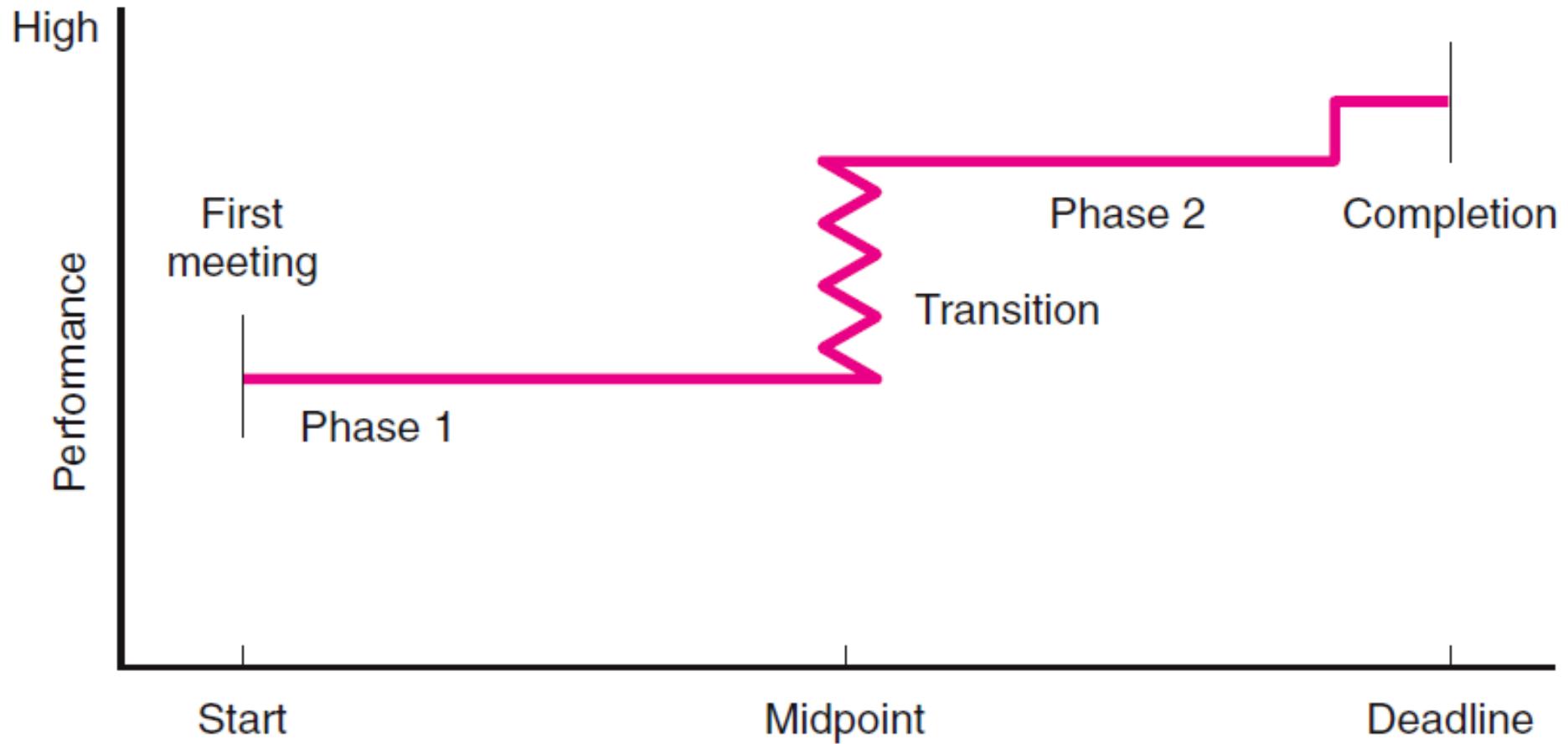
The Five-Stage Team Development Model



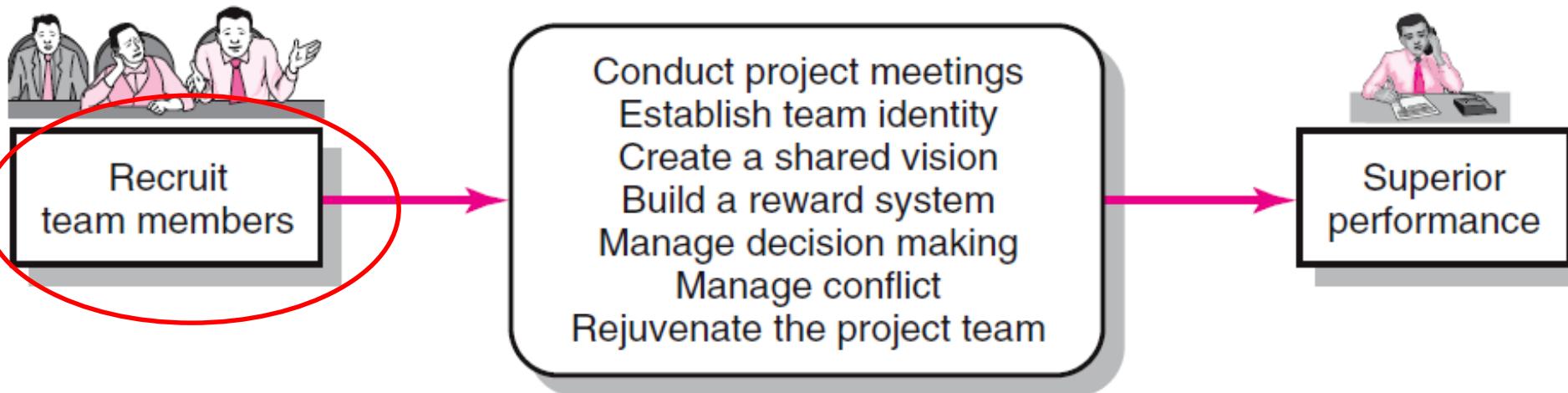
Conditions Favoring Development of High Performance Project Teams

- **Ten or fewer team members**
- **Voluntary team membership**
- **Continuous service on the team**
- **Full-time assignment to the team**
- **An organization culture of cooperation and trust**
- **Members report only to the project manager**
- **All relevant functional areas are represented on the team**
- **The project has a compelling objective**
- **Members are in speaking distance of each other**

The Punctuated Equilibrium Model of Group Development



Creating a High-Performance Project Team



Building High-Performance Project Teams

■ Recruiting Project Members

□ Factors affecting recruiting

- Importance of the project
- Management structure used to complete the project

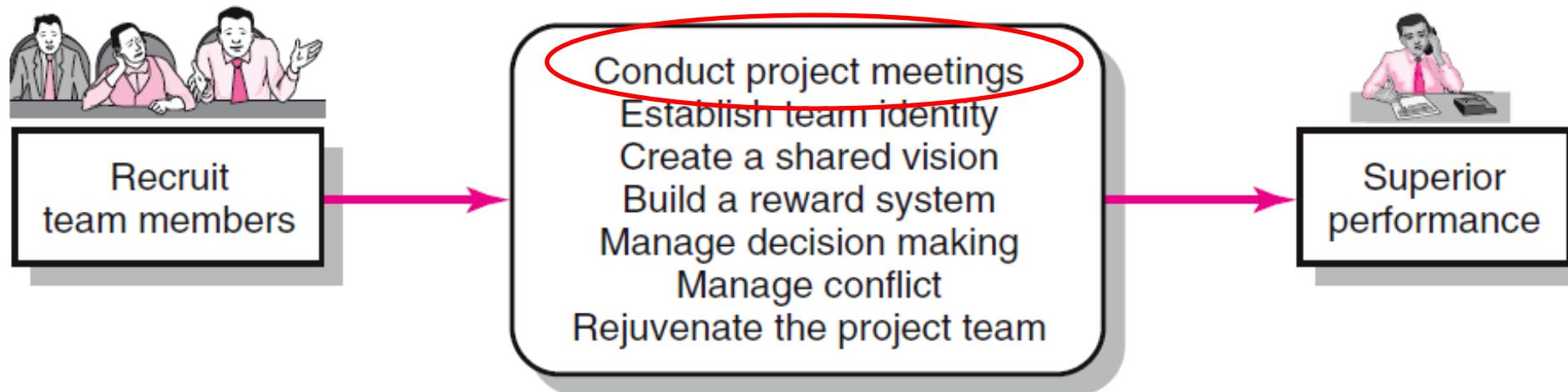
□ How to recruit?

- Ask for volunteers

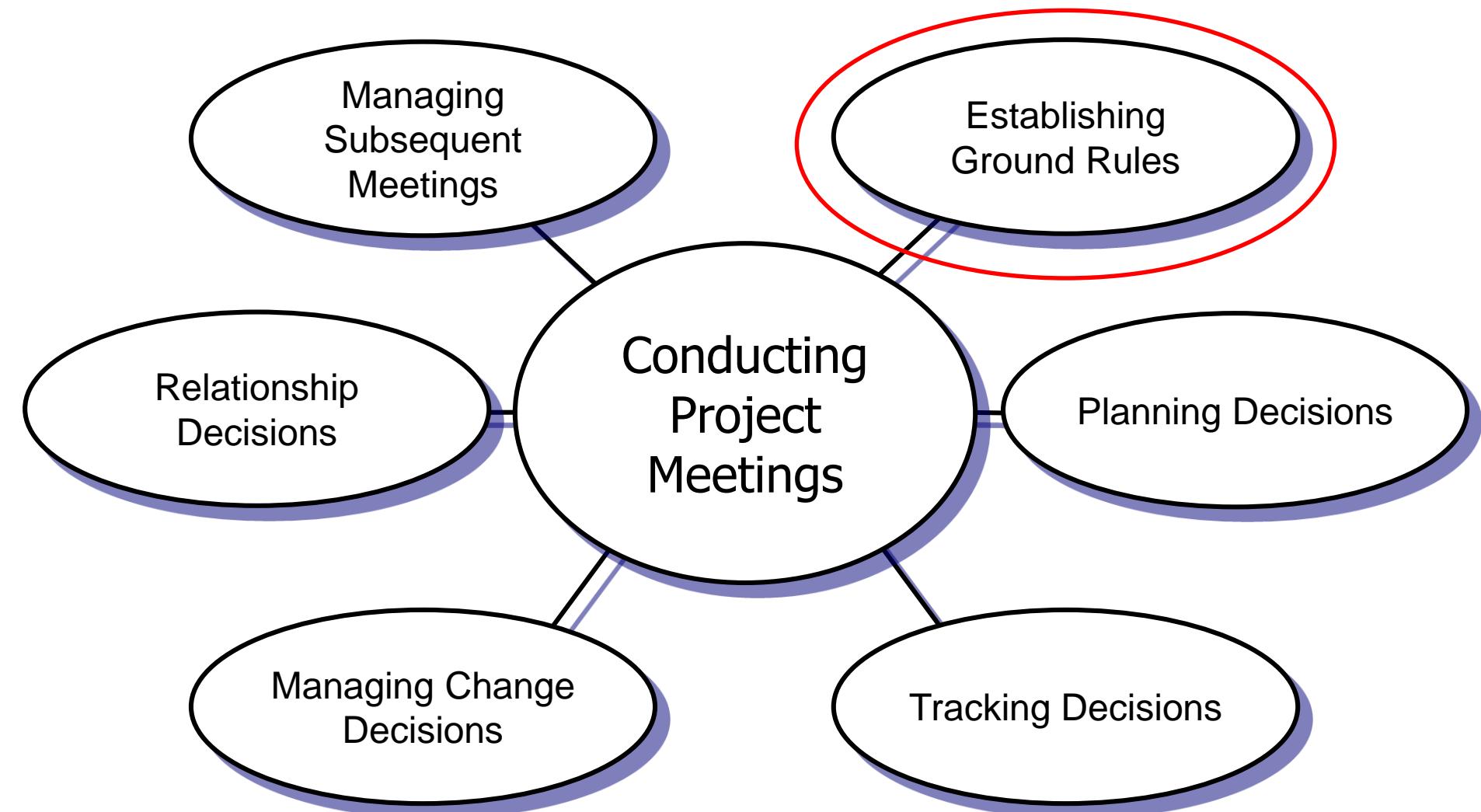
□ Who to recruit?

- Problem-solving ability
- Availability
- Technological expertise
- Credibility
- Political connections
- Ambition, initiative, and energy

Creating a High-Performance Project Team



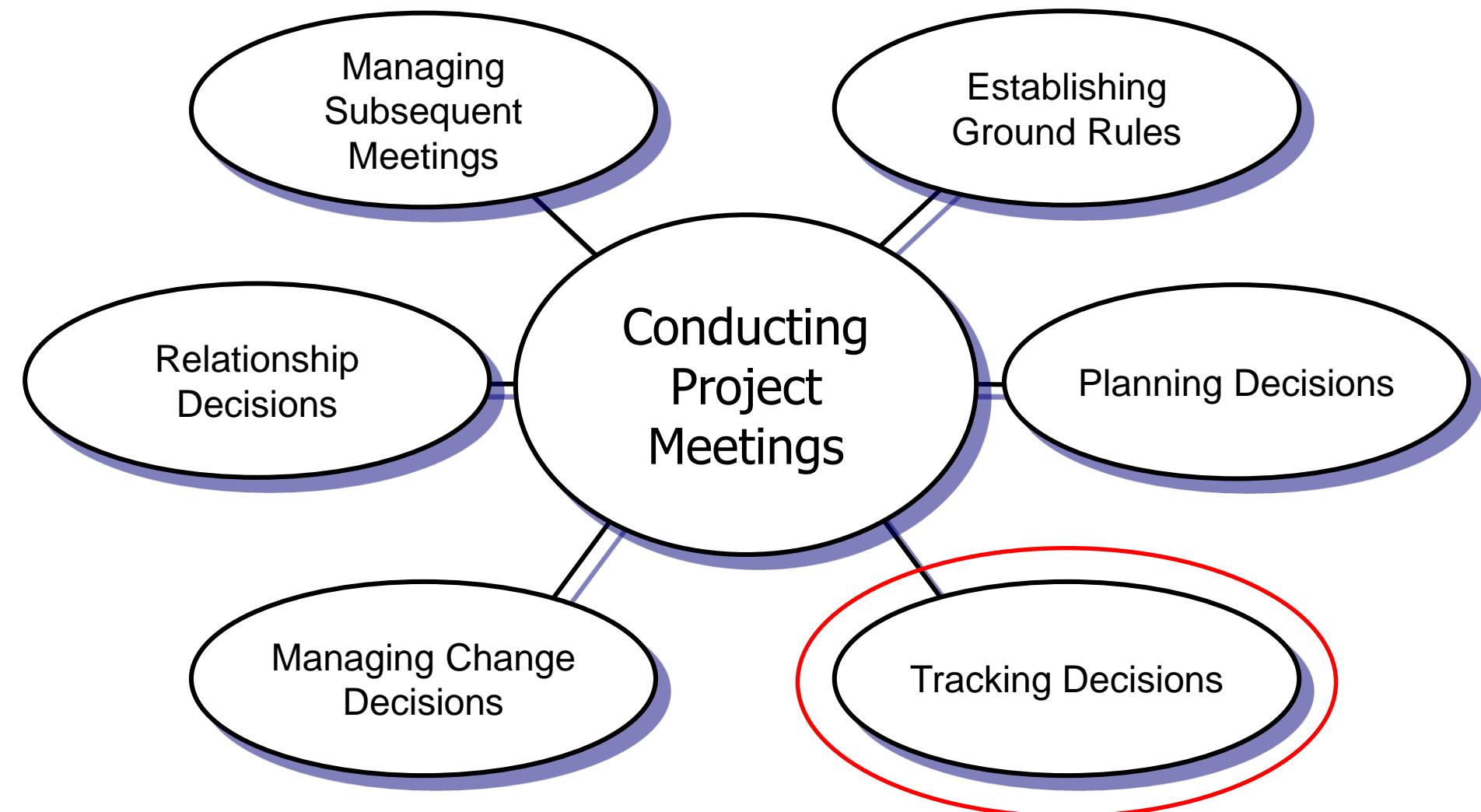
Project Team Meetings

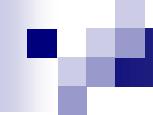


Establishing Ground rules

- ***Planning Decisions***
- How will the project plan be developed?
- What tools will be used to support the project?
- Will a specific project management software package be used? If so, which one?
- Who will enter the planning information?
- What are the specific roles and responsibilities of all the participants?
- Who needs to be informed of decisions? How will they be kept informed?
- What are the relative importance of cost, time, and performance?
- What are the deliverables of the project planning process?
- What format is appropriate for each deliverable?
- Who will approve and sign off at the completion of each deliverable?
- Who receives each deliverable?

Project Team Meetings

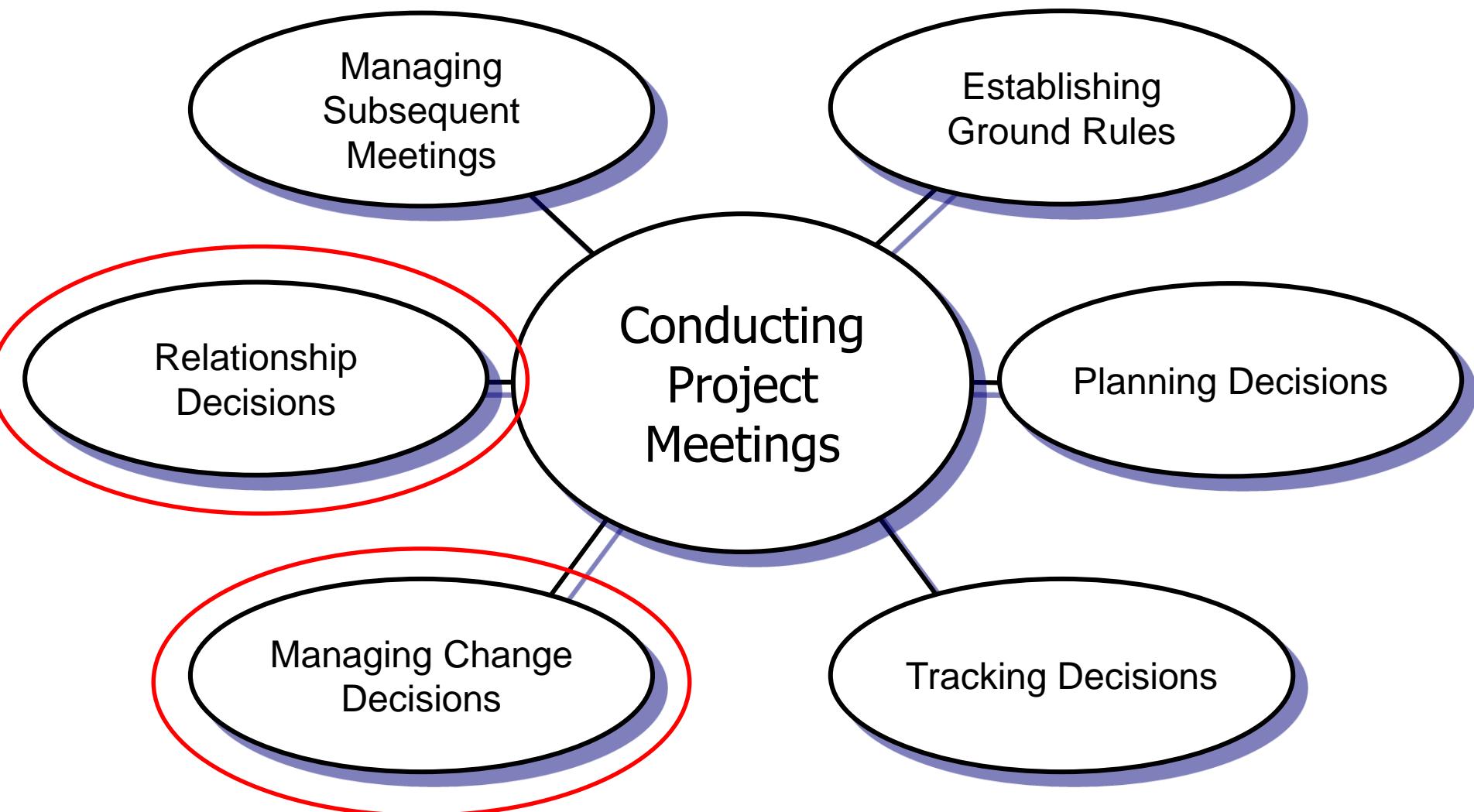




Tracking Decisions

- How will progress be assessed?
- At what level of detail will the project be tracked?
- How will team members get data from each other?
- How often will they get this data?
- Who will generate and distribute reports?
- Who needs to be kept informed about project progress, and how will they be informed?
- What content/format is appropriate for each audience?
- Meetings
 - Where will meetings be located?
 - What kind of meetings will be held?
 - Who will “run” these meetings?
- How will agendas be produced?
- How will information be recorded?

Project Team Meetings



Other decisions

■ ***Managing Change Decisions***

- How will changes be instituted?
- Who will have change approval authority?
- How will plan changes be documented and evaluated?

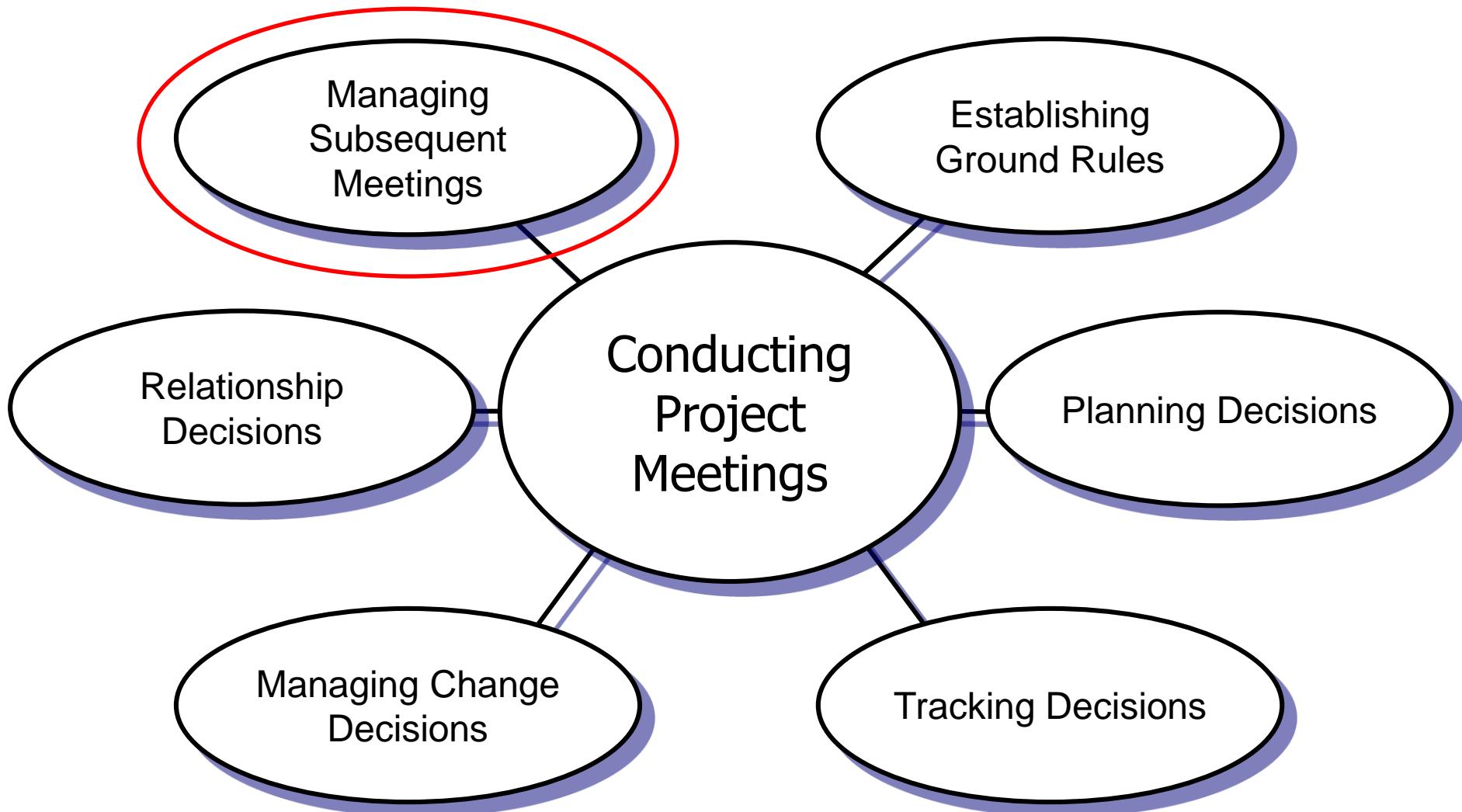
■ ***Relationship Decisions***

- What department or organizations will the team need to interact with during the project?
- What are the roles and responsibilities of each organization (reviewer, approver, creator, user)?
- How will all involved parties be kept informed of deliverables, schedule dates, expectations, etc.?
- How will the team members communicate among themselves?
- What information will and won't be exchanged?

Team interaction norms

- **Confidentiality** is maintained; no information is shared outside the team unless all agree to it.
- It is acceptable to **be in trouble**, but it is **not acceptable to surprise others**. Tell others immediately when deadlines or milestones will not be reached.
- There is **zero tolerance for bullying** a way through a problem or an issue.
- **Agree to disagree**, but when a decision has been made, regardless of personal feelings, move forward.
- **Respect outsiders**, and do not flaunt one's position on the project team.
- **Hard work** does not get in the way of **having fun**.

Project Team Meetings



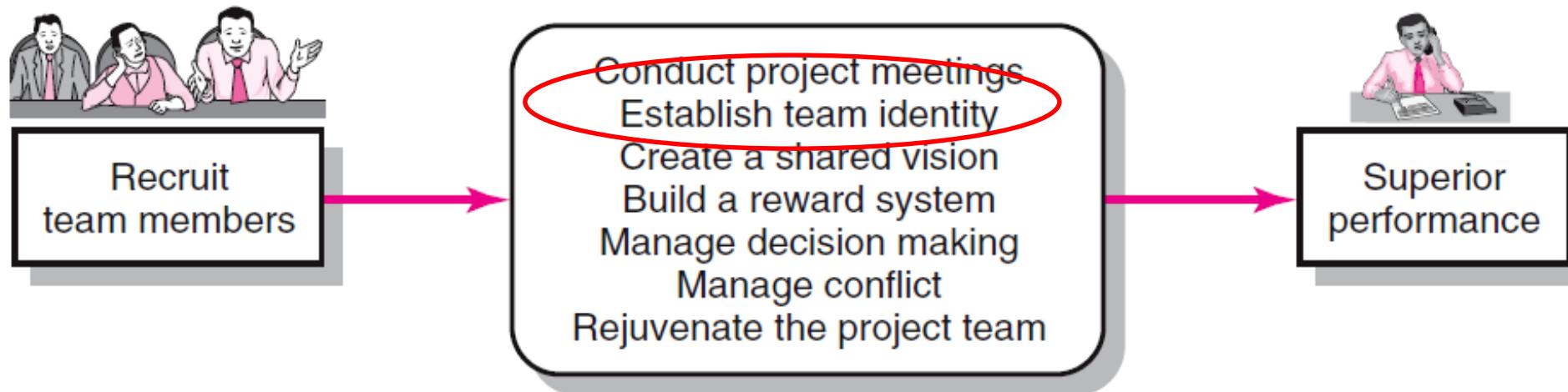
Managing Subsequent Project Meetings

- Start meetings on time regardless of whether everyone is present.
- Prepare and distribute an agenda prior to the meeting.
- Identify an adjournment time.
- Periodically take time to review how effective previous meetings have been.
- Solicit recommendations and implement changes.
- Assign good recordkeeping.
- Review the agenda before beginning, and tentatively allocate time for each item.

Managing Subsequent Project Meetings

- Prioritize issues so that adjustments can be made given time constraints.
- Encourage active participation of all members by asking questions instead of making statements.
- Summarize decisions, and review assignments for the next meeting.
- Prepare and distribute a summary of the meeting to appropriate people.
- Recognize accomplishments and positive behavior.

Creating a High-Performance Project Team



Establishing a Team Identity

**Effective Use
of Meetings**

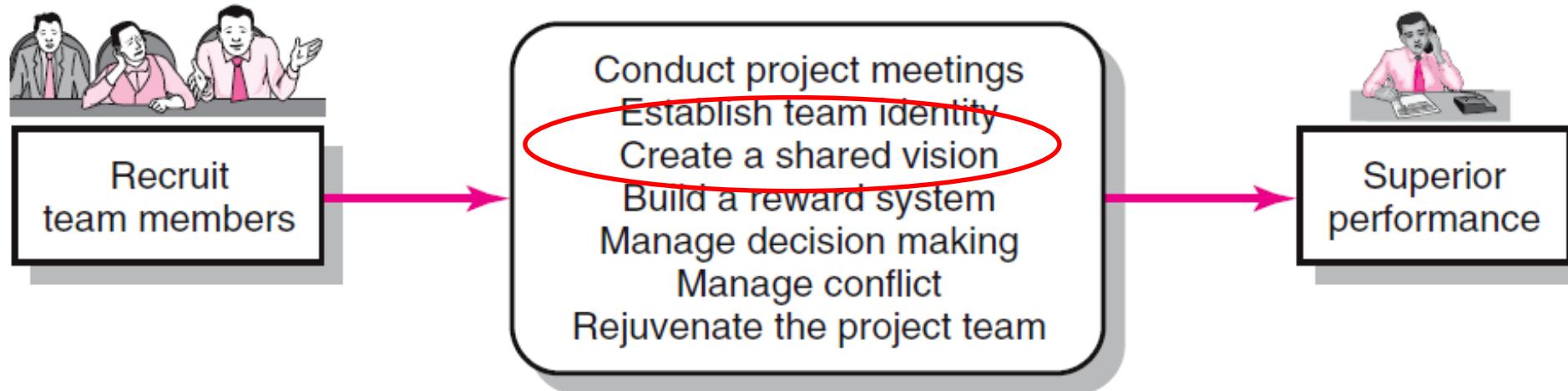
**Co-location of
team members**

**Creation of project
team name**

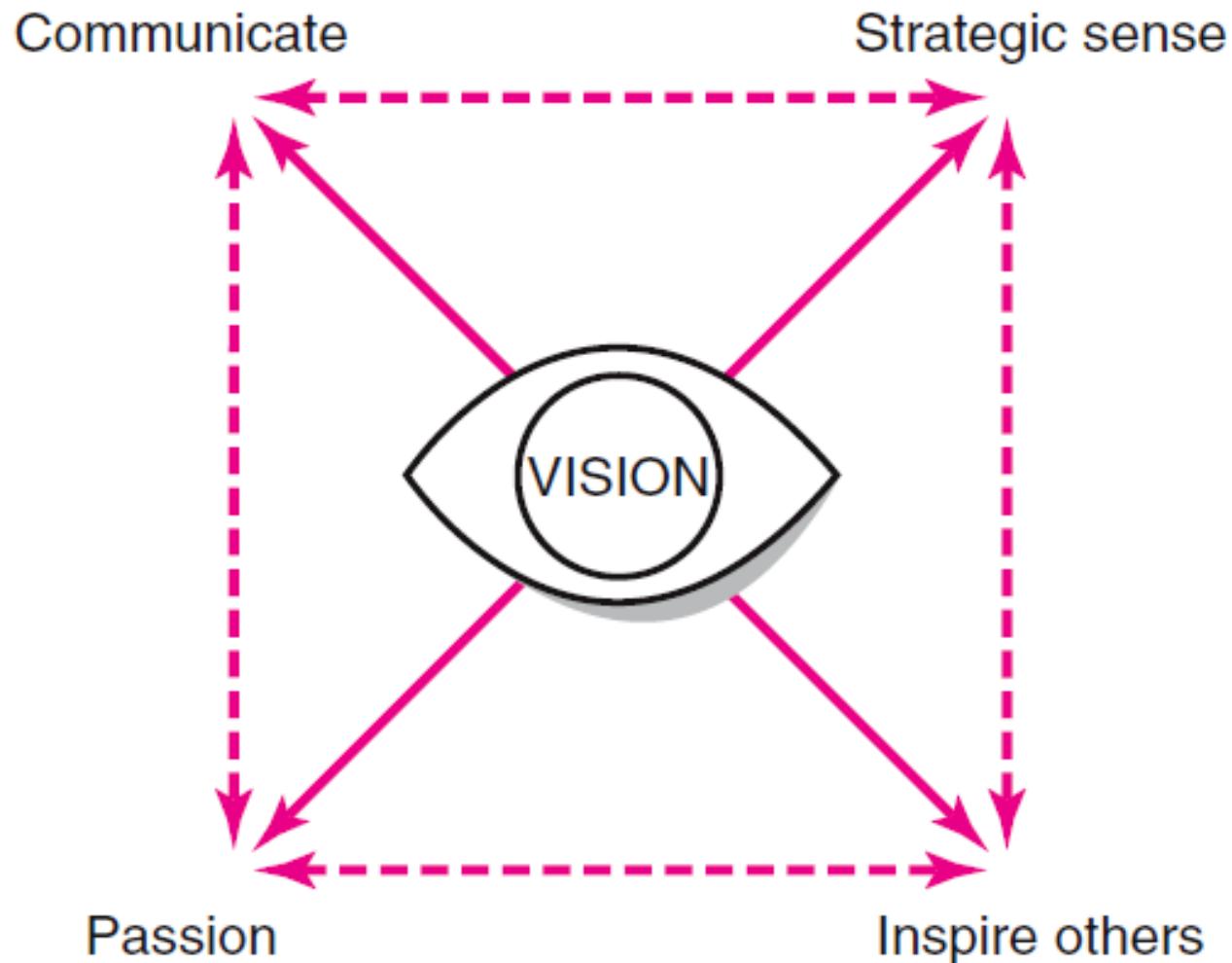
Team rituals



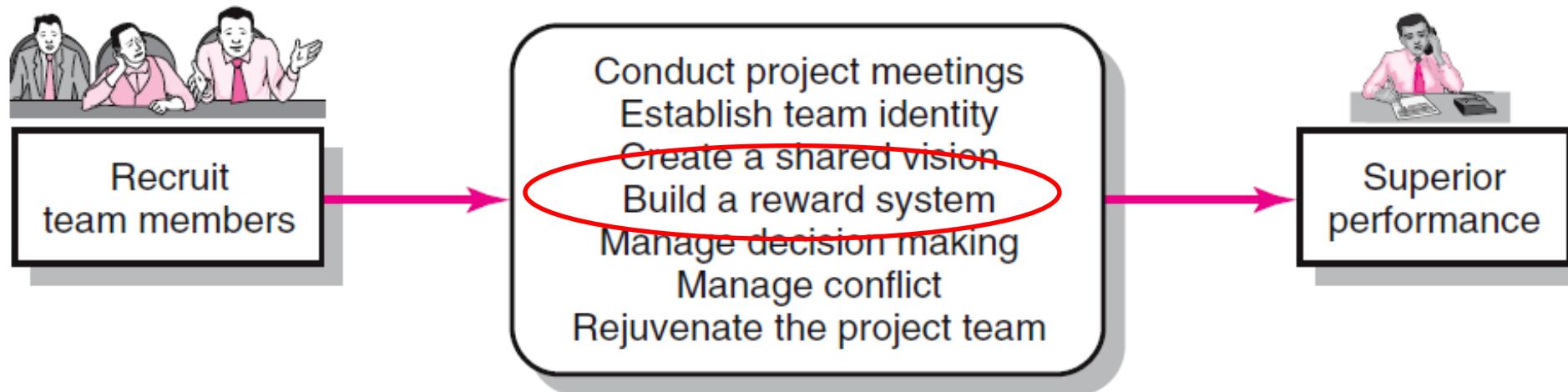
Creating a High-Performance Project Team



Requirements for an Effective Shared Vision



Creating a High-Performance Project Team

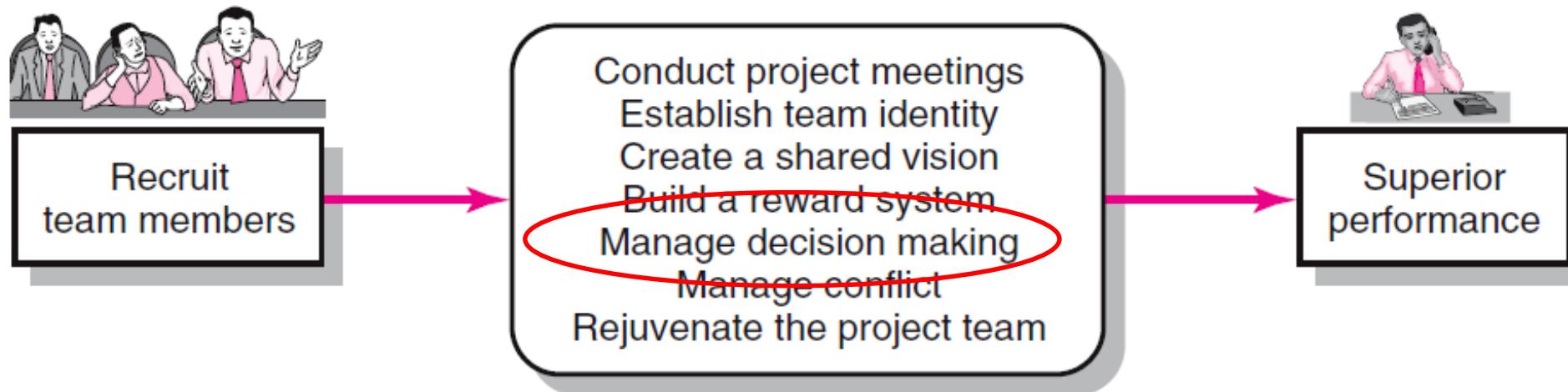


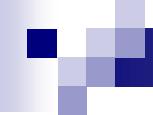
Managing Project Reward Systems

■ Group Rewards

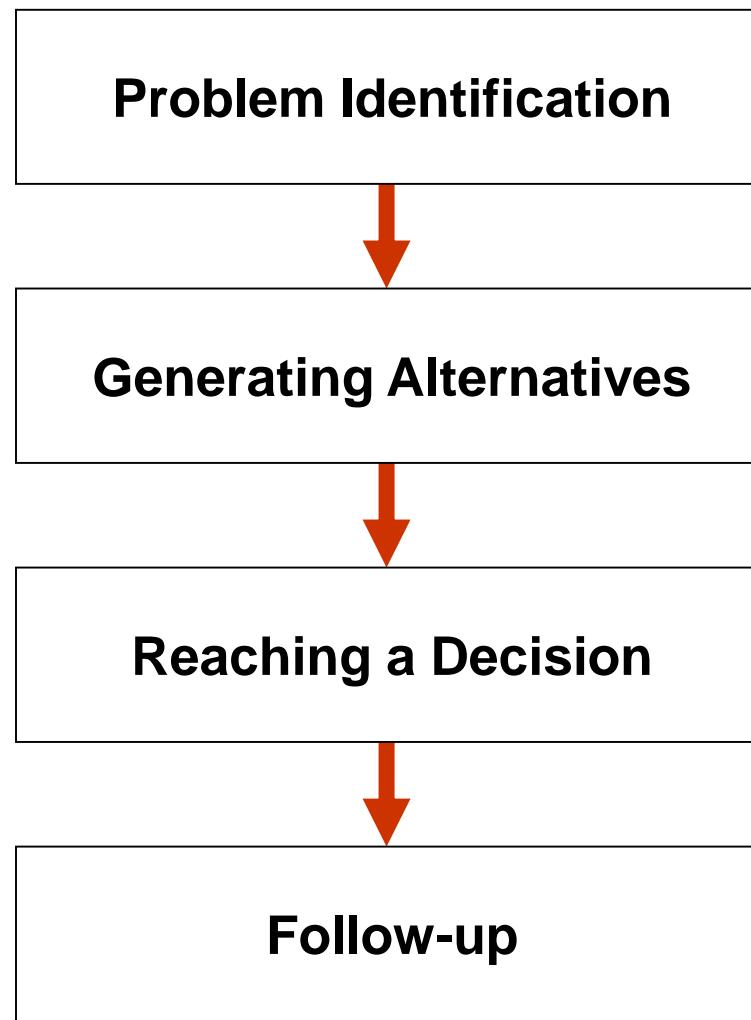
- Who gets what as an individual reward?
- How to make the reward have lasting significance?
- How to recognize individual performance?
 - Letters of recommendation
 - Public recognition for outstanding work
 - Desirable job assignments
 - Increased personal flexibility

Creating a High-Performance Project Team

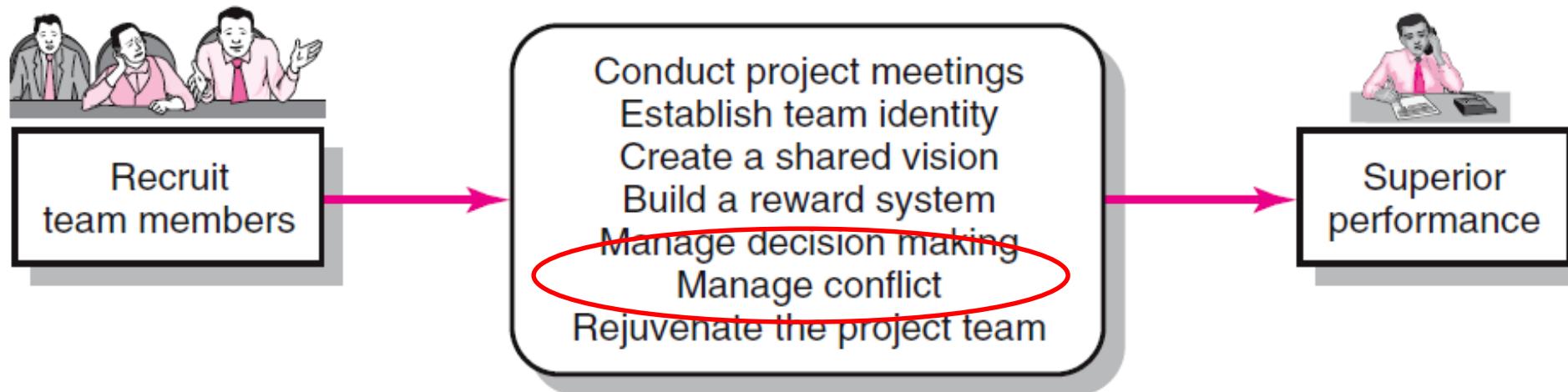




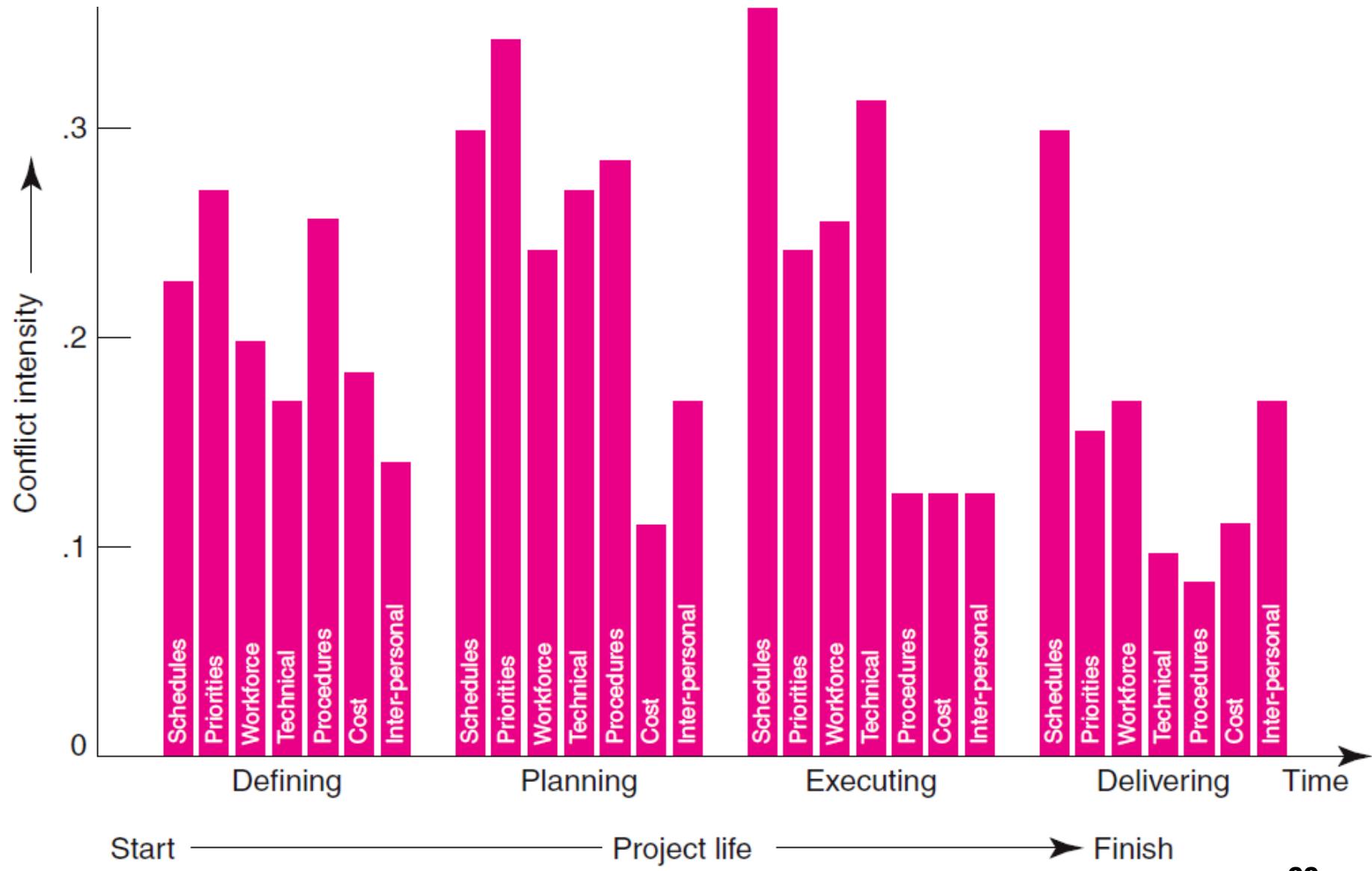
Orchestrating the Decision Making Process

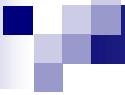


Creating a High-Performance Project Team



Conflict Intensity Over the Project Life Cycle





Managing Conflict within the Project Team

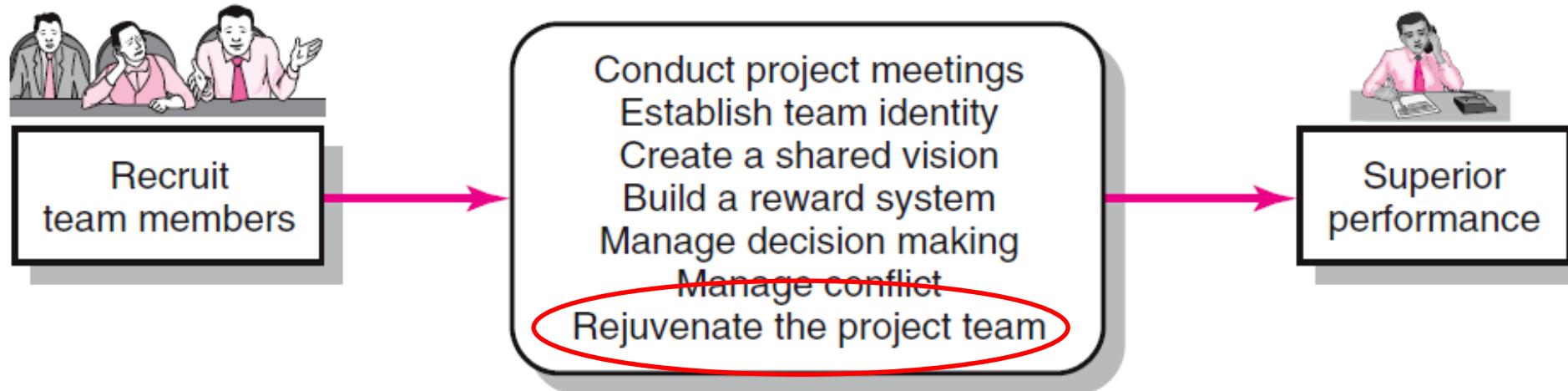
■ Encouraging Functional Conflict

- Encourage dissent by asking tough questions.
- Bring in people with different points of view.
- Designate someone to be a devil's advocate.
- Ask the team to consider an unthinkable alternative

■ Managing Dysfunctional Conflict

- Mediate the conflict.
- Arbitrate the conflict.
- Control the conflict.
- Accept the conflict.
- Eliminate the conflict.

Creating a High-Performance Project Team



Rejuvenating the Project Team

■ Informal Techniques

- Institute new rituals.
- Take an off-site break as a team from the project.
- View an inspiration message or movie.
- Have the project sponsor give a talk.

■ Formal Techniques

- Hold a team building session facilitated by an outsider to clarify ownership issues affecting performance.
- Engage in an outside activity that provides an intense common experience to promote social development of the team.

Managing Virtual Project Teams

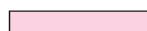
- Challenges:
 - Developing trust
 - Exchange of social information.
 - Set clear roles for each team member.
 - Developing effective patterns of communication.
 - Keep team members informed on how the overall project is going.
 - Don't let team members vanish.
 - Establish a code of conduct to avoid delays.
 - Establish clear norms and protocols for surfacing assumptions and conflicts.
 - Share the pain.

24-Hour Global Clock

United States (East Coast)	Australia	Scotland	Comments
12 midnight	2 PM	5 AM	
1 AM	3 PM	6 AM	
2 AM	4 PM	7 AM	
3 AM	5 PM	8 AM	
4 AM	6 PM	9 AM	Australia handoff for off-shift review
5 AM	7 PM	10 AM	
6 AM	8 PM	11 AM	3-way conferencing window (primary)
7 AM	9 PM	12 noon	3-way conferencing window (primary)
8 AM	10 PM	1 PM	3-way conferencing window (primary)
9 AM	11 PM	2 PM	
10 AM	12 midnight	3 PM	
11AM	1 AM	4 PM	
12 noon	2 AM	5 PM	Scotland handoff for off-shift review
1 PM	3 AM	6 PM	
2 PM	4 AM	7 PM	
3 PM	5 AM	8 PM	
4 PM	6 AM	9 PM	3-way conferencing window (secondary)
5 PM	7 AM	10 PM	3-way conferencing window (secondary)
6 PM	8 AM	11 PM	U.S. handoff for off-shift review
7 PM	9 AM	12 midnight	
8 PM	10 AM	1 AM	
9 PM	11 AM	2 AM	
10 PM	12 noon	3 AM	
11 PM	1 PM	4 AM	
12 midnight	2PM	5 AM	



Prime time

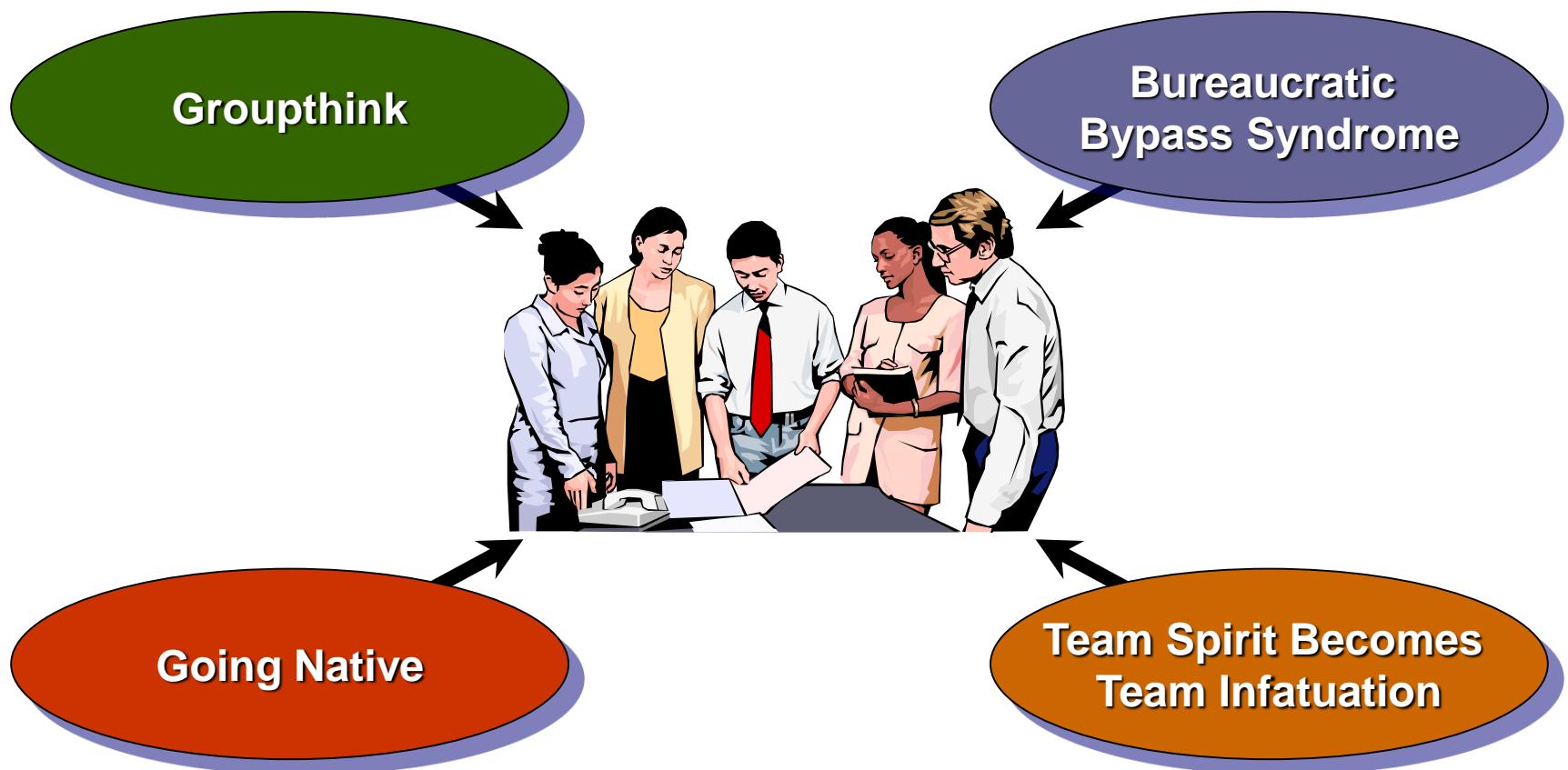


Secondary time



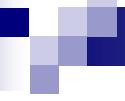
Downtime

Project Team Pitfalls



Wrap-up

- People means
 - Team members
 - Stakeholders
 - Clients
 - Partners
- Management is about handling complexity
- Leadership is about handling change



Quiz time

- Let's switch over to Moodle

Discussion topic for next time

- Any combination of the previous topics so far
- Plus any feedback is welcome (i.e. what you liked, what you did not like, what would you change about this class)

HAVE A GREAT HOLIDAY AND AN AWESOME NEW YEAR!!



Change Management

Lecture 11

...recap::People management

- Managing versus Leading a Project
- Managing—coping with **complexity**
 - Formulate plans and objectives
 - Monitor results
 - Take corrective action
 - Expedite activities
 - Solve technical problems
 - Serve as pacemaker
 - Make tradeoffs among time, costs, and project scope

...recap:: Managing versus Leading a Project

- Leading—coping with **change**
 - Recognize the need to change to keep the project on track
 - Initiate change
 - Provide direction and motivation
 - Innovate and adapt as necessary
 - Integrate assigned resources

..recap:: Network of Stakeholders

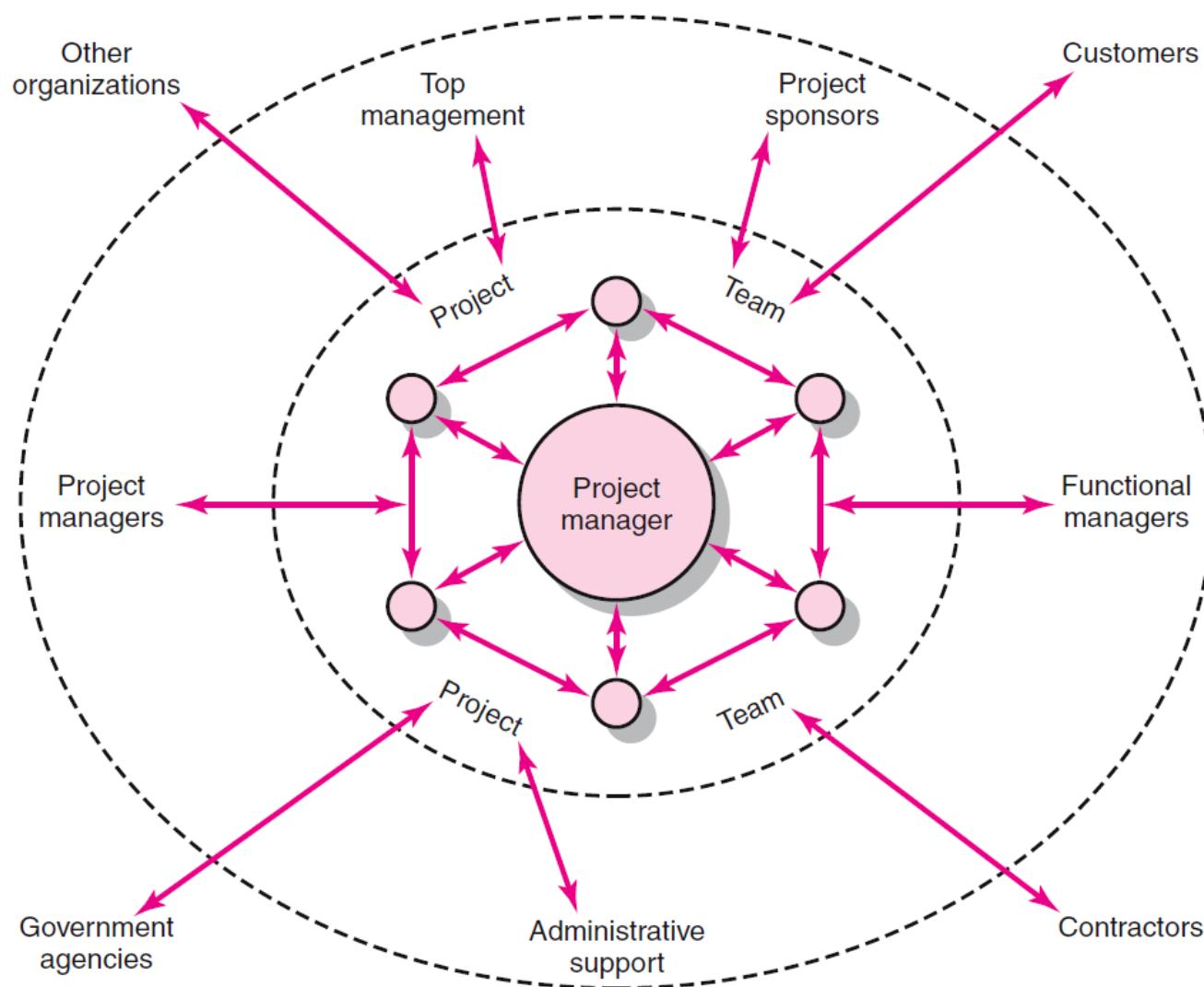
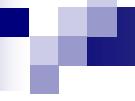


FIGURE 10.1



...recap:: Characteristics of Effective Project Managers

- Initiate contact with key players.
- Anticipate potential problems.
- Provide encouragement.
- Reinforce the objectives and vision of the project.
- Intervene to resolve conflicts and prevent stalemates.

..recap:: Managing Upward Relations

- Project Success = Top Management Support
 - Appropriate budgets
 - Responsiveness to unexpected needs
 - A clear signal to the organization of the importance of cooperation
- Motivating the Project Team
 - Influence top management in favor of the team:
 - Rescind unreasonable demands
 - Provide additional resources
 - Recognize the accomplishments of team members

..recap:: Leading by Example

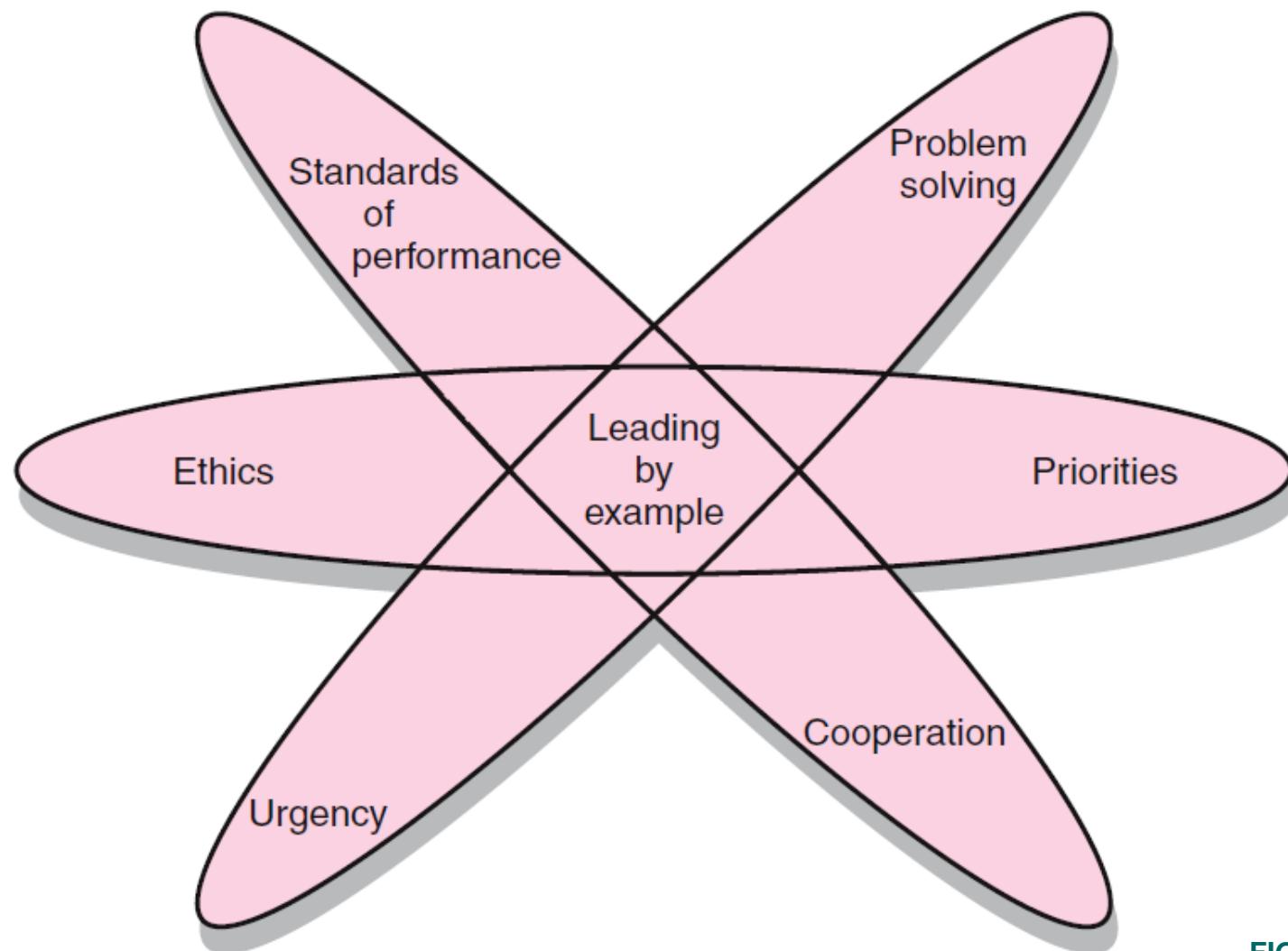
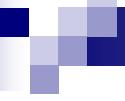


FIGURE 10.4

....recap:: Trust

- **Ethic => Trust**
- **Character** (Does he want the right thing?)
 - ⇒ Consistency (predictable)
 - ⇒ Openness
- **Competence** (Does he know the right thing to do?)
 - Tasks
 - Interpersonal
 - Organizational

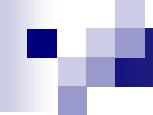


Today's topic

- Change Management

References

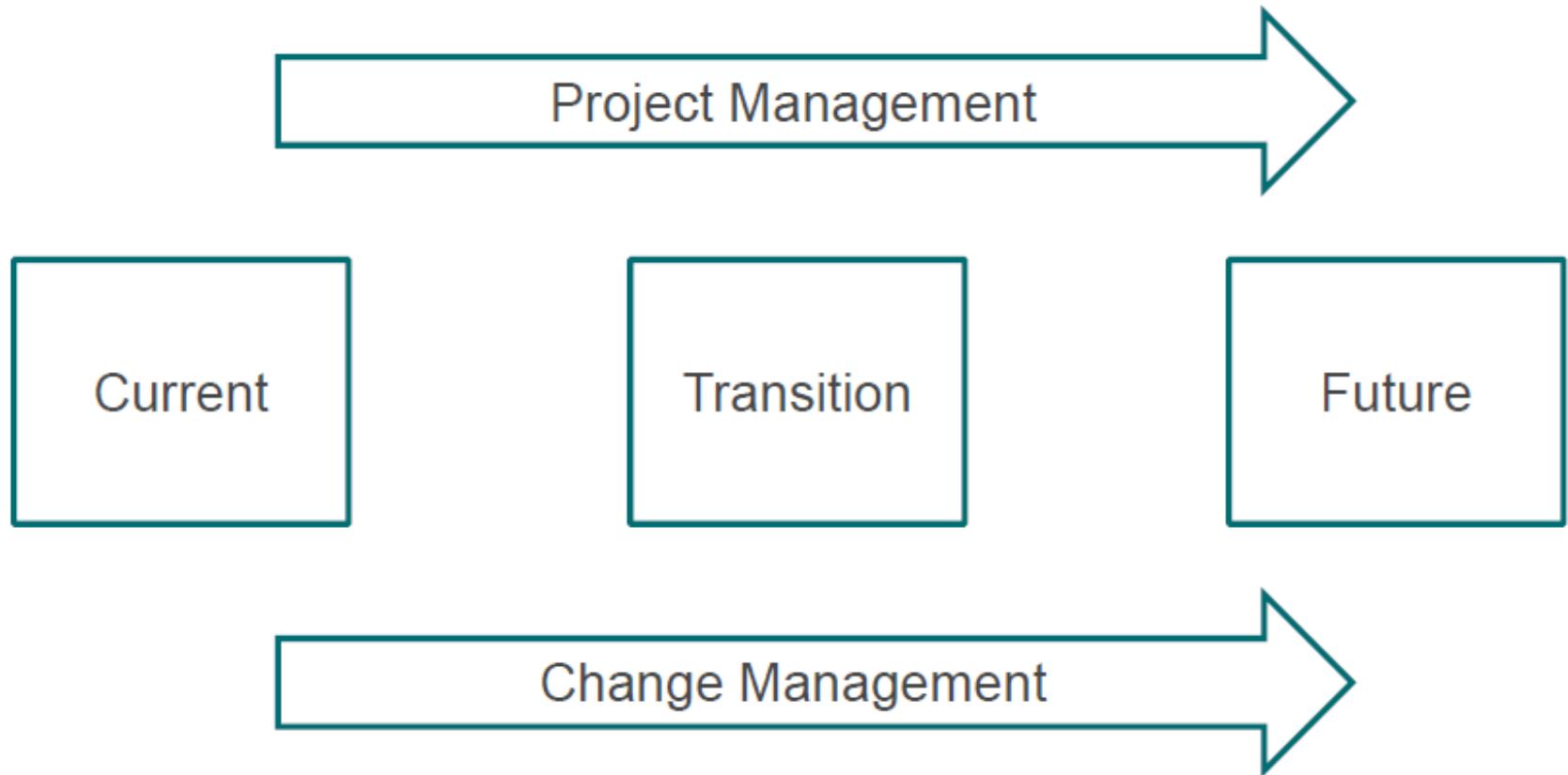
- PMBOK
- CMMI
- Change Management within Project Management: An Integrated Structural Business Process Approach, 2012, BCS Hellenic Section
- Prosci® Change Management Overview, 2014



"It must be considered that there is nothing more **difficult** to carry out nor more **doubtful of success** nor more **dangerous** to handle than to **initiate a new order of things.**"

Machiavelli (1446-1507)

Project management and Change management



PMBOK - Integrated Change Control

- Part of the **Project Integration Management** Knowledge Area:
 - Develop Project Charter, Develop Preliminary Project Scope Statement, Develop Project Management Plan, Direct and Manage Project Execution, Monitor and Control Project Work, **Integrated Change Control**, and Close Project

PM Process Group

- Integrated Change Control (ICC) and all processes in the Project Integration Area are coordinated within the **Project Management Process Groups**

Definition

- *ICC is the process necessary for controlling factors that create changes to make sure those changes are beneficial, determining whether a change has occurred, and managing the approved changes, including when they occur.*
- ICC process is performed throughout the project, from project initiation through project closure.

Activities [1]

- Identify that a change needs to occur or has occurred.
- Influence the factors that circumvent integrated change control so that **only approved changes are implemented**.
- Review and **approve/reject** requested changes.

Activities [2]

- Manage the approved changes when and as they occur, by **regulating the flow of requested changes**.
- Maintain the integrity of baselines by **releasing only approved changes** for incorporation into project products or services, and maintain their related configuration and planning documentation.

Activities[3]

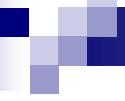
- Review and **approve recommended** corrective and preventive **actions**.
- Control and update the scope, cost, budget, schedule and quality requirements based upon approved changes, by **coordinating changes** across the entire project.

Activities[4]

- **Document the complete impact of requested changes.**
- Validate defect repair.
- Control project quality to standards based on quality reports.

Inputs

- Project Management Plan
- Requested Changes
- Work Performance Information
- Recommended Preventive Actions
- Recommended Corrective Actions
- Recommended Defect Repair
- Deliverables



Tools and Techniques

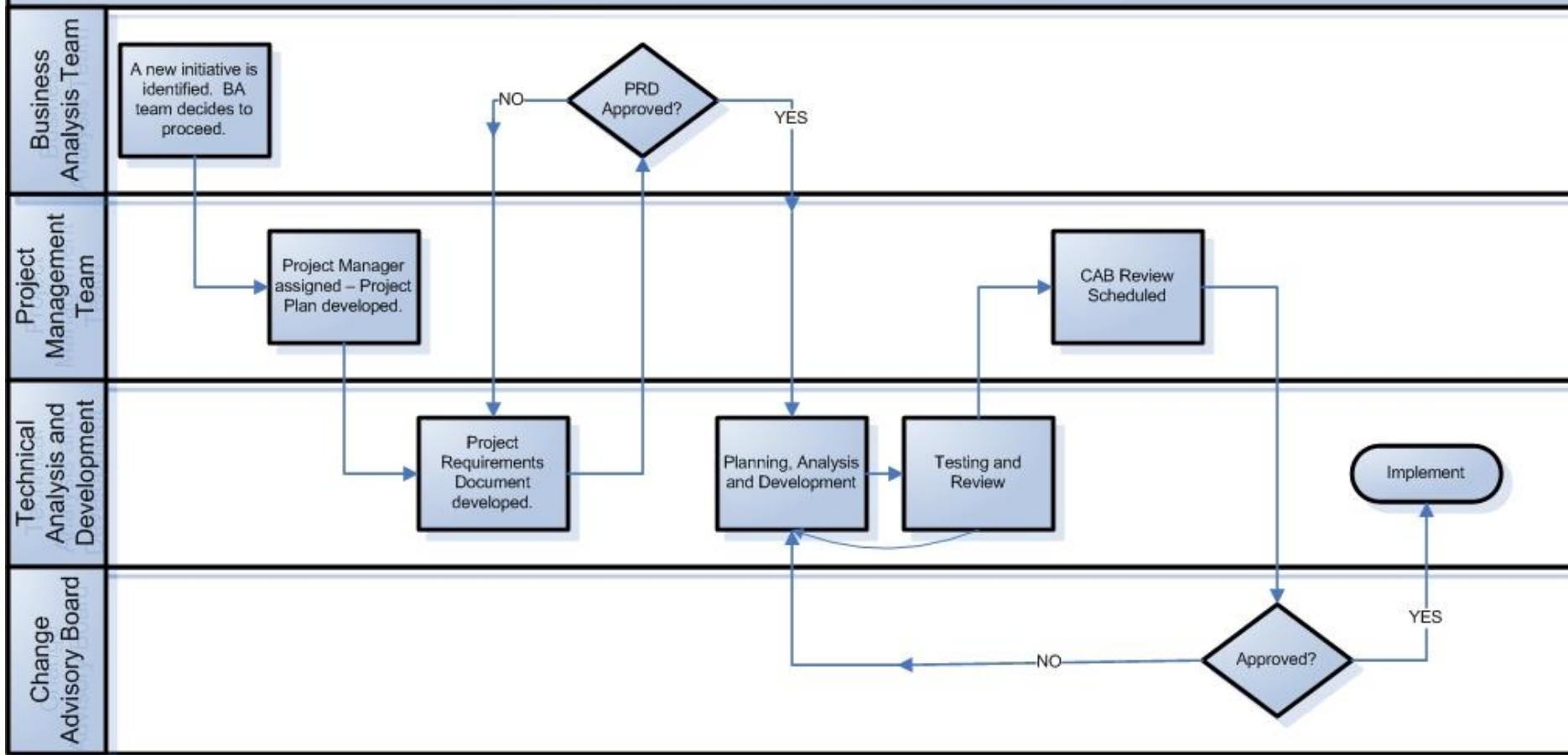
- Project Management Methodology
- Project Management Information System
- Expert Judgment

Outputs

- Approved Change Requests
- Rejected Change Requests
- Project Management Plan (Updates)
- Project Scope Statement (Updates)
- Approved Corrective Actions
- Approved Preventive Actions
- Approved Defect Repair
- Deliverables

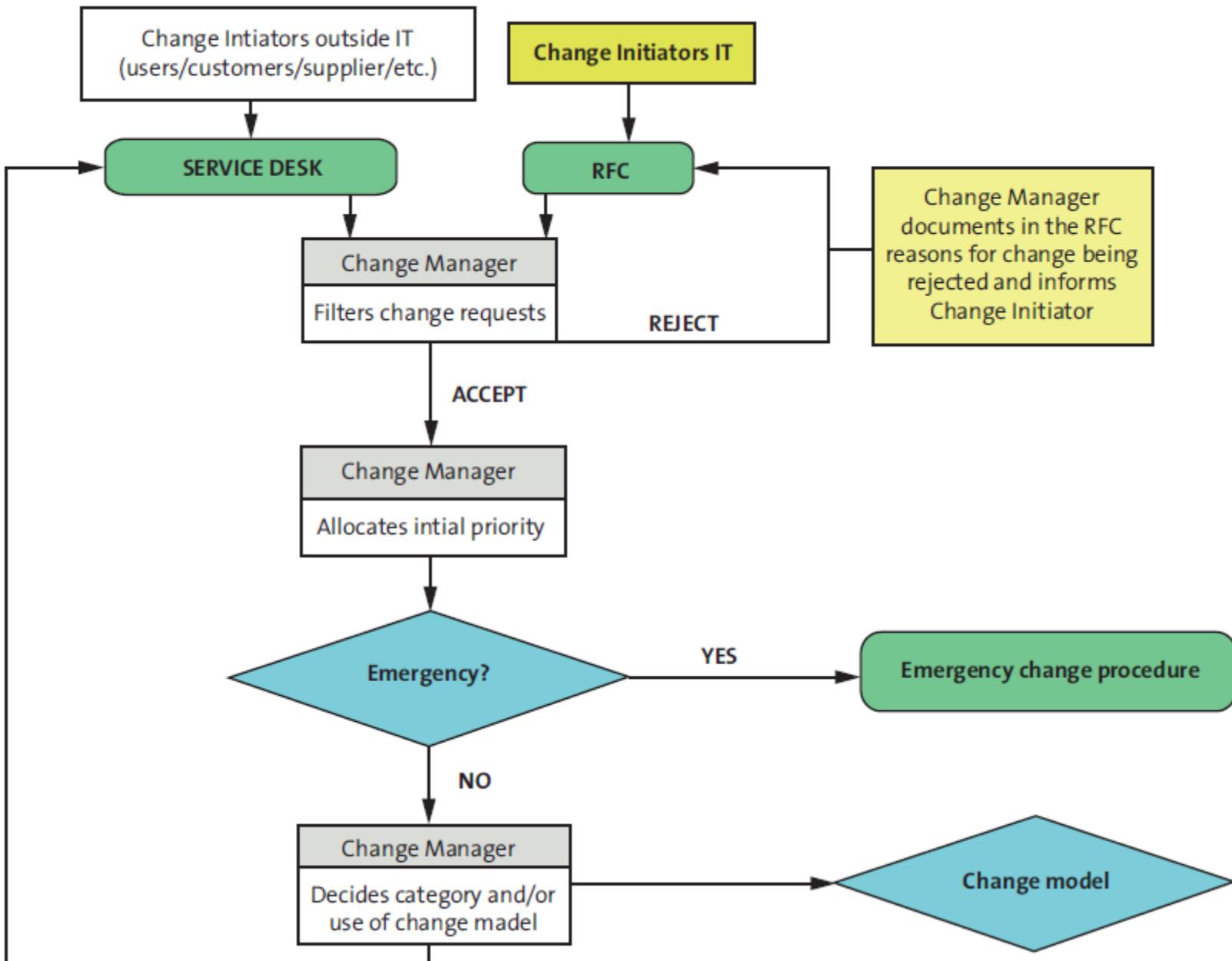
Example

How a Technology Change Moves from Inception to Implementation



Examples

- **Procedures** should be subject to Change Management.
 - Ex. a change in system backup scheduling, must go through Change Management.
- **Standard periodic maintenance** should be preapproved.
 - Ex. If normally a server reboots on Sunday morning at 2:00 AM, it is not necessary to submit an RFC each time, but that process must be approved in advance.
- **Ad-hoc maintenance** must adhere to the CMP.



Change Request

- Change information should include the following:
 - The **reason/business justification** for the change
 - Why the change is needed – implications of **not implementing** the change
 - Known **risks or impact** to the business of implementing the change
 - Required **resources**

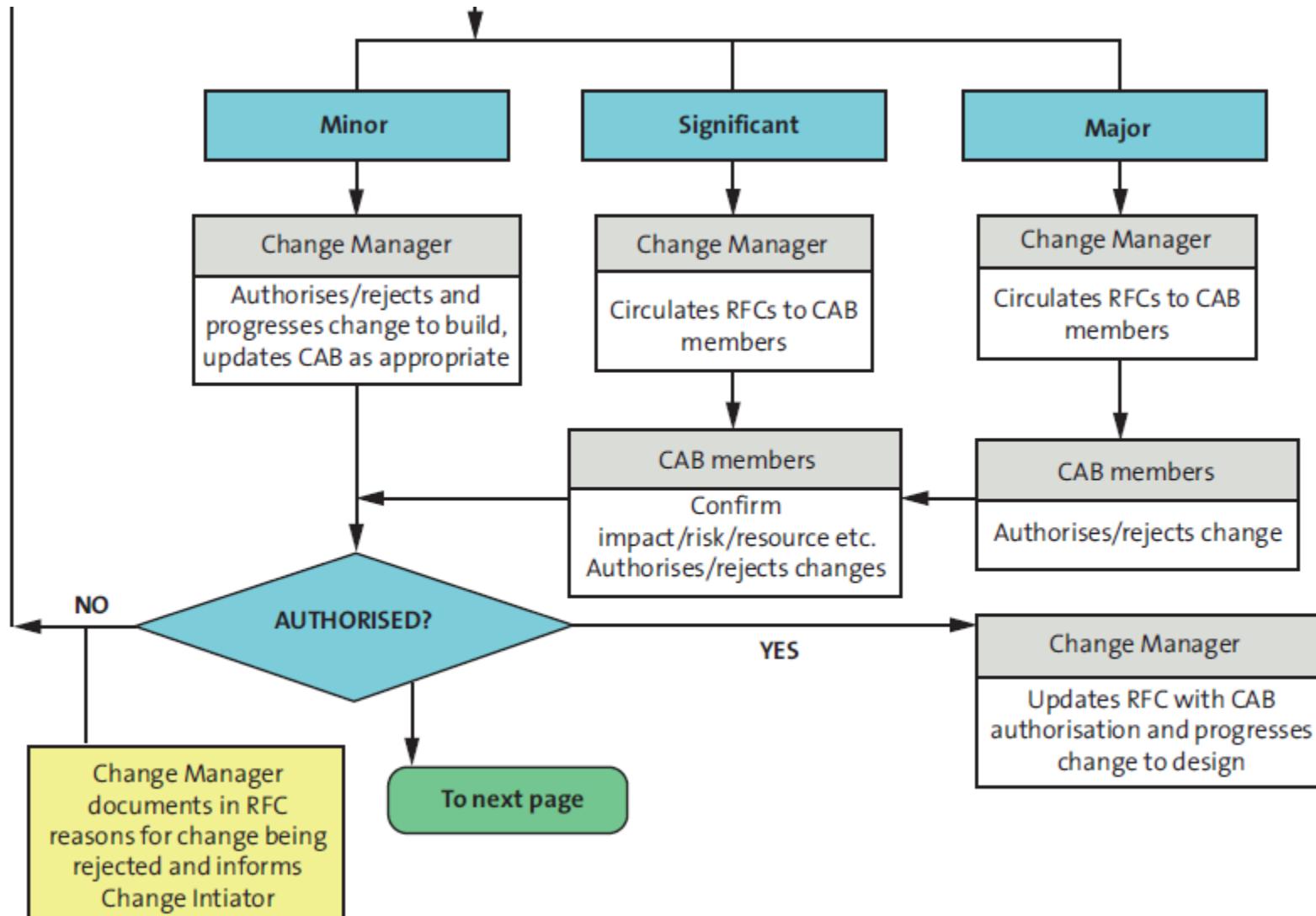
Change attributes

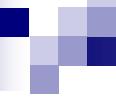
■ Priority

- Emergency** – needs doing immediately
(emergency change process)
- High** – needs doing within 48 hours.
- Medium** – needs doing within five days.
- Low** – needs doing by the indicated date

Change attributes [2]

- Category
 - **Standard** change – using a procedure – pre-authorized
 - **IT change** model – using a procedure – may need some level of authorization
 - **Minor** change – authorized by Change Manager alone (low risk and low impact to the business)
 - **Significant** change – authorized by a CAB (medium risk and/or medium impact to the business)
 - **Major** change – authorized by a CAB (senior level) (high risk and/or high impact to the business)





Authorization of a change by the Change Advisory Board (CAB)

■ A standard CAB agenda

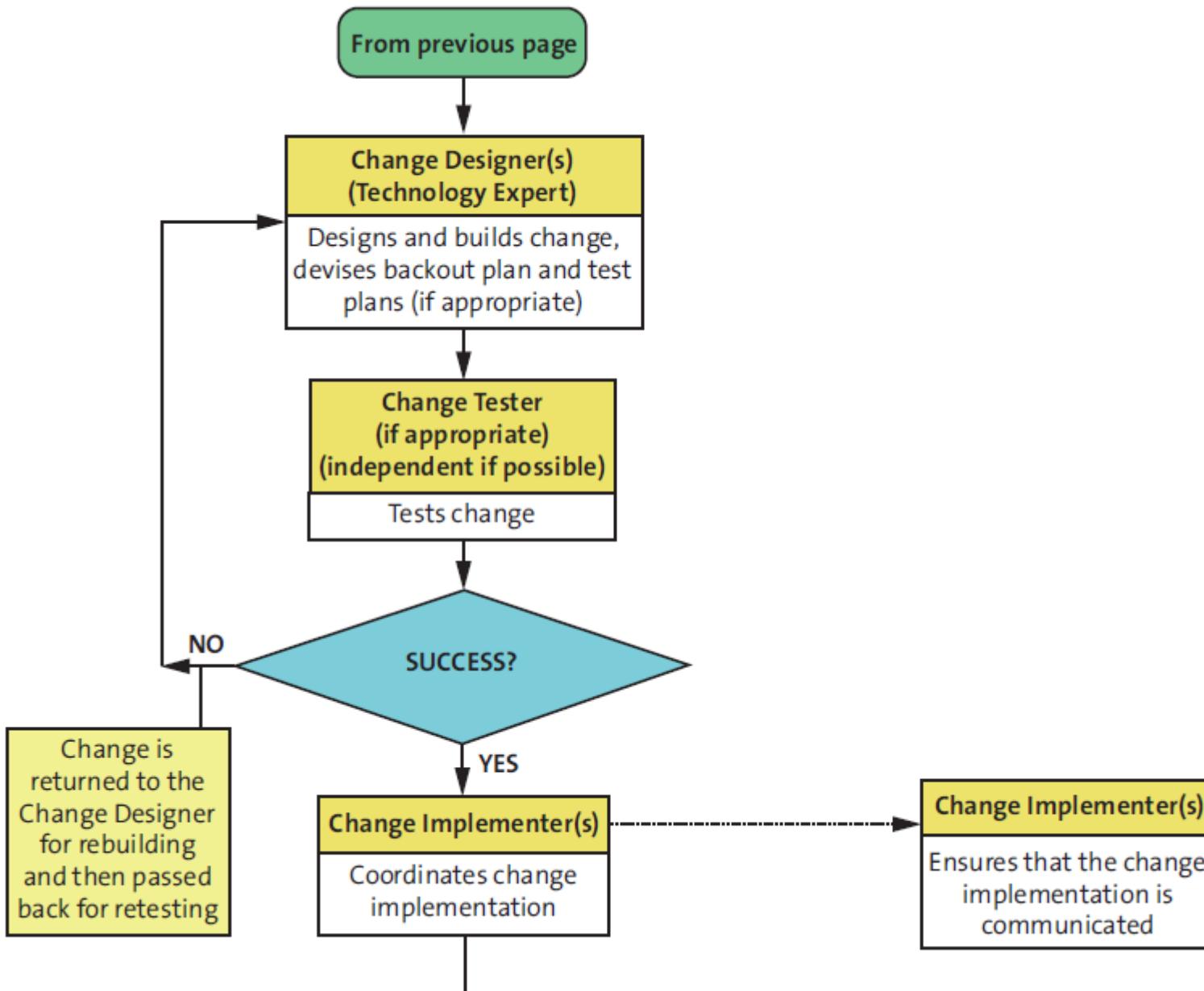
- Failed changes
- Backed out changes
- RFCs to be assessed by CAB members
- RFCs that have been assessed by CAB members
- Implemented changes are reviewed
- The change management process – including any amendments made to the process
- Change management successes for the period under discussion

CAB considerations for each change (prior to authorization)

- **Impact** assessment (on the business)
- **Risk** assessment (on the business)
- Effect upon the **infrastructure and customer service**, as defined in the SLA, and upon the **capacity and performance, reliability and resilience, contingency plans, and security**
- Impact on **other services** that run on the same infrastructure (or on software development projects)
- **Resource** assessment – the IT, business and other resources required to implement the change, covering the likely costs, the number and availability of people required, the elapsed time, and any new infrastructure elements required

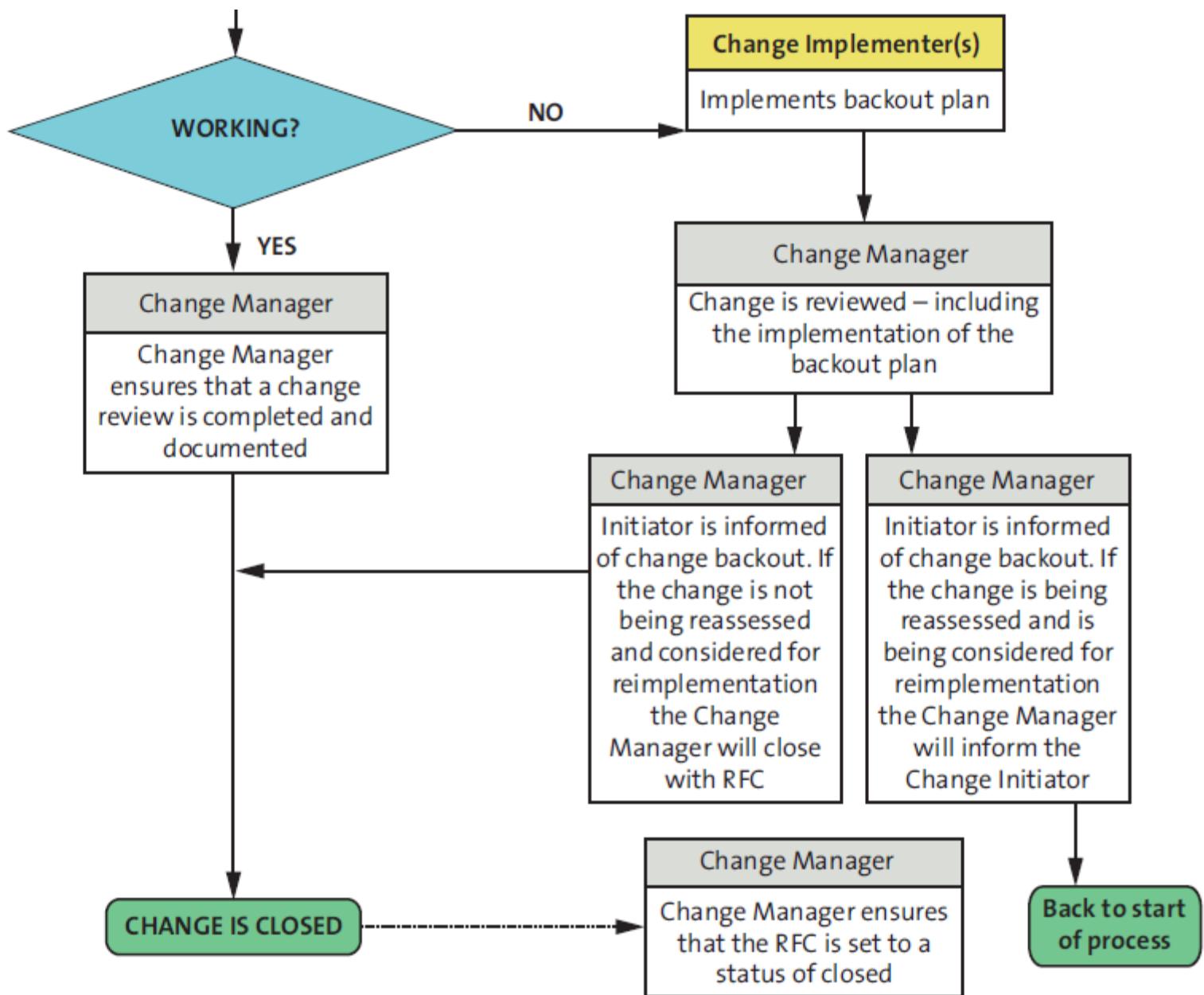
CAB considerations for each change (prior to authorization) [2]

- The impact on non-IT infrastructures within the organization
- **Effect/risk/impact of not implementing the change**
- Technical capability and technical approval
- Financial approval (if required)
- Third party/supplier involvement in the implementation of the change
- Business approval (if required)
- Review/assessment of the change priority



Activities of change building

- building a new production module
- creating a new version of one or more software modules
- purchasing equipment or services externally
- preparing a hardware modification
- producing new or amended documentation showing the components of the change build
- devising a **back-out plan**
- devising testing requirements, as appropriate
- documenting required resources for the change implementation



Change review

- The change has had **the desired effect** and met its objectives
- Users and **customers are content** with the results, or to identify any shortcomings
- There have been **no unexpected or undesirable side effects** to functionality, availability, capacity/performance, security, maintainability etc.
- The **resources** used to implement the change **were as planned**
- The implementation **plan worked correctly** (so include comments from the implementers)
- The change was implemented **on time and to cost**
- The **back-out plan functioned correctly**, if the back-out plan was implemented

Examples of changes



Ad hoc processes → Documented and managed processes

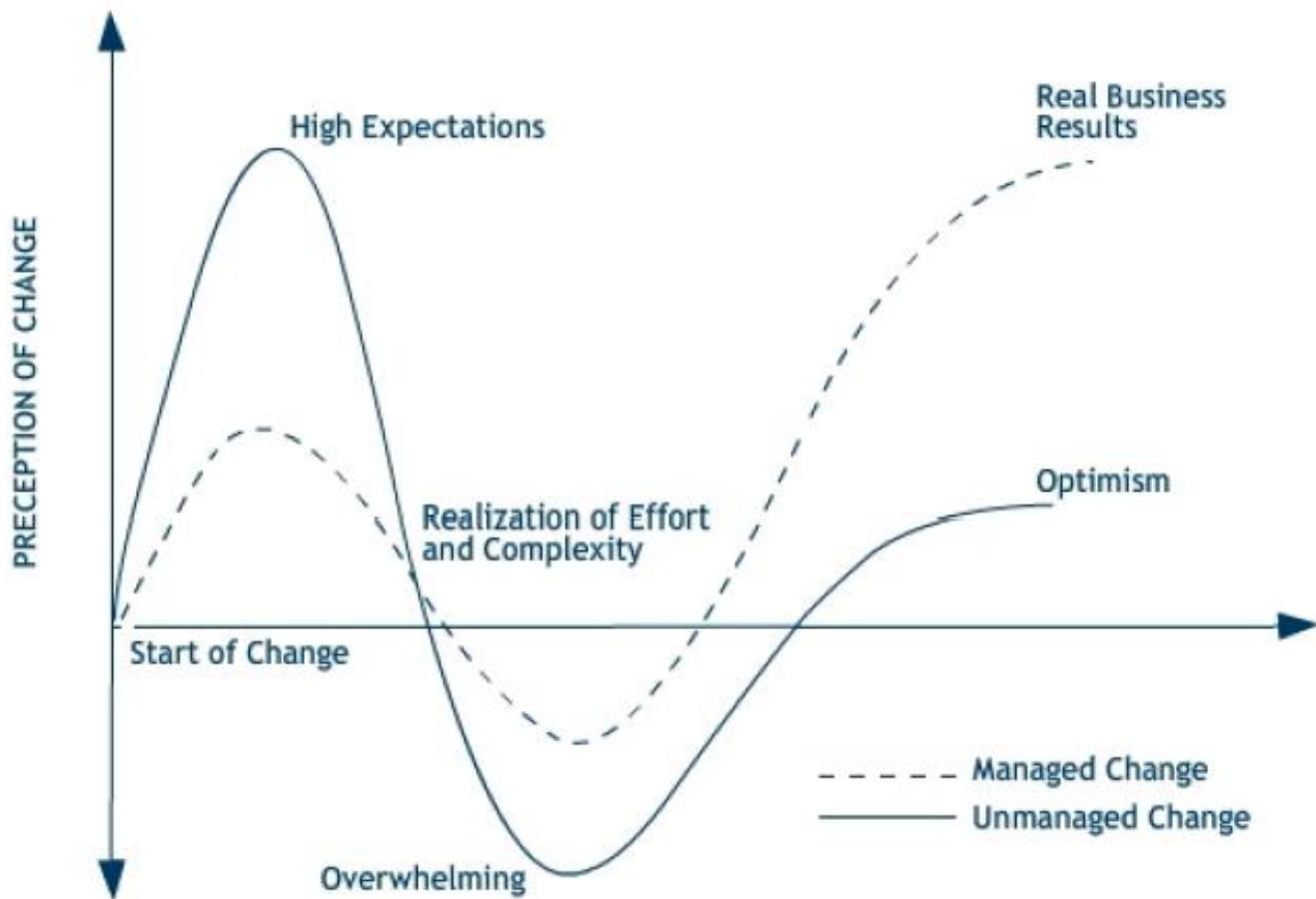
Multiple, legacy systems → One integrated database

Generalists in the call center → Specialists in the call center

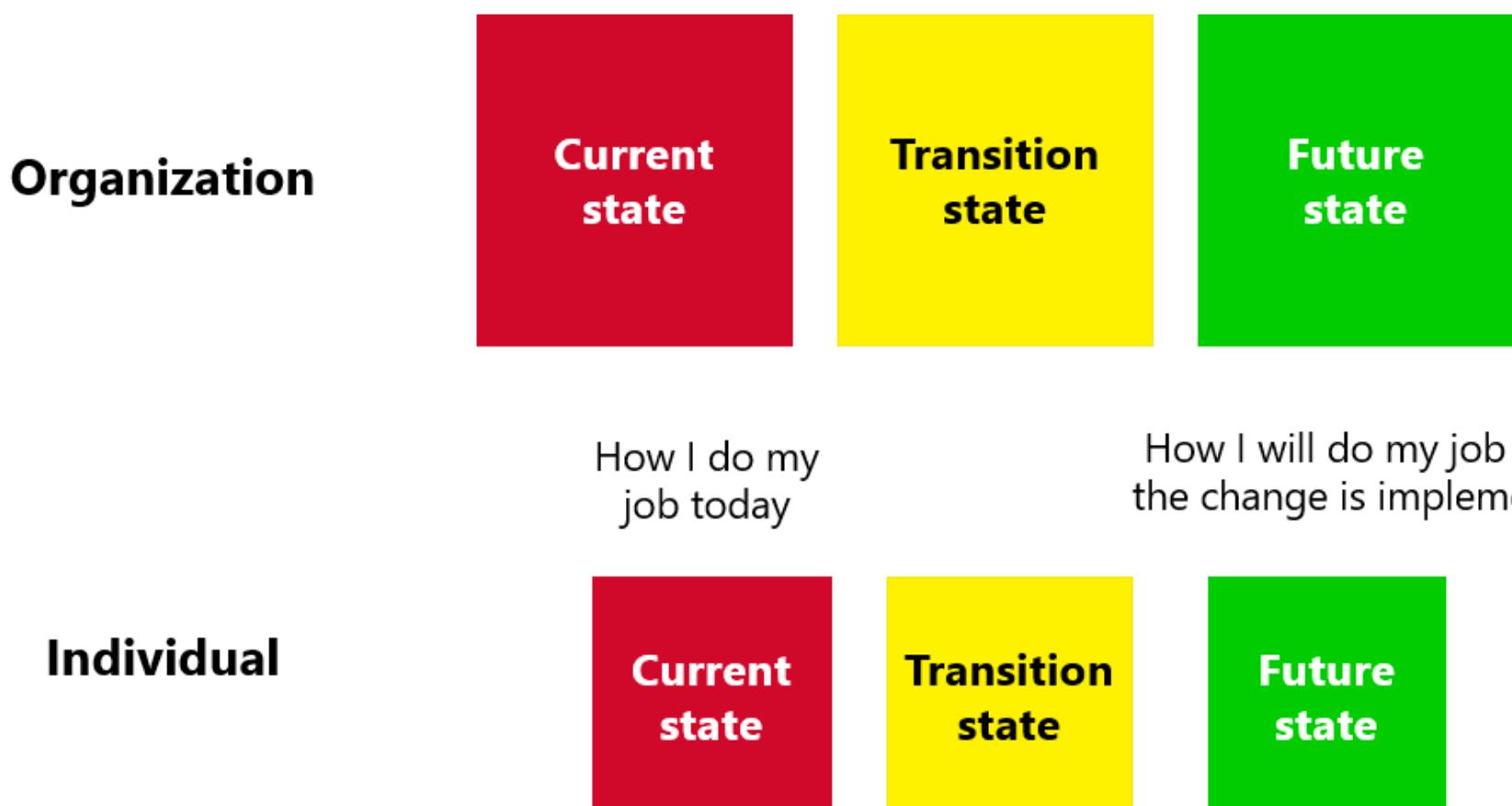
No web interface for suppliers → Supplier website integrated into supply chain

Two different companies → Merged organization

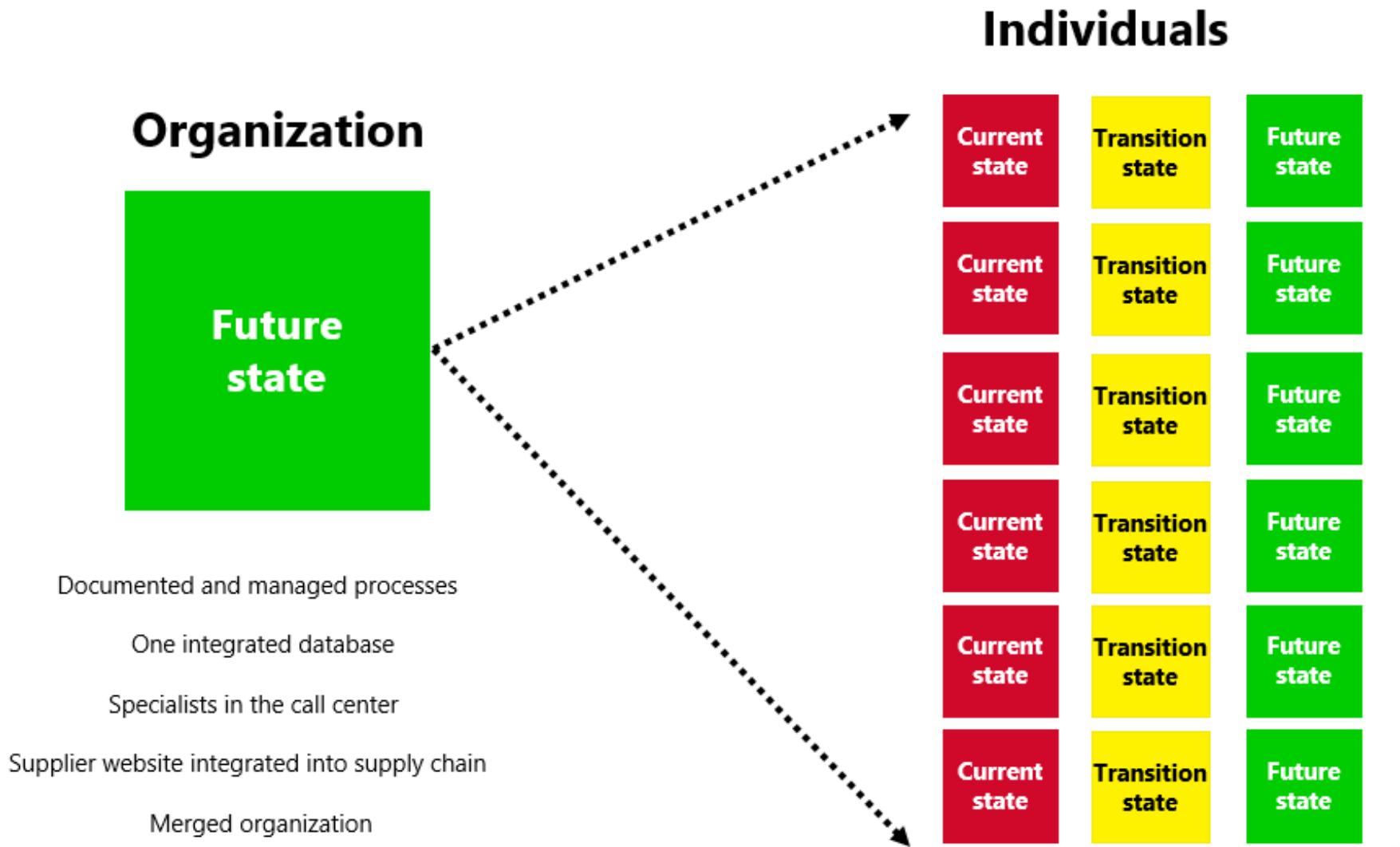
Project change curve



People side of Change management



Organization future





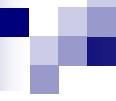
Not managing people change

- Lower productivity
- Passive resistance
- Active resistance
- Turnover of valued employees
- Disinterest in the current or future state
- Arguing about the need for change
- More people taking sick days or not showing up



Not managing people change

- Changes not fully implemented
- People finding work-arounds
- People revert to the old way of doing things
- The change being totally scrapped
- Divides are created between ‘us’ and ‘them’



Change Management Models

- Lewin's Change Management Model
- McKinsey 7-S Model
- Kotter's 8 Step Change Model
- ADKAR Model (Prosci)

Lewin's Change Management Model

- 1. Unfreeze** – overcome the active effort to resist change through motivation.
- 2. Transition** –a transition period may last for some time. Adequate leadership and reassurance is necessary
- 3. Refreeze** – After change has been accepted and successfully implemented, the company becomes stable again, and staff refreezes under the new guidelines.

McKinsey 7-S Model

7 factors that operate as collective agent of change:

1. Shared values
2. Strategy
3. Structure
4. Systems
5. Style
6. Staff
7. Skills

Kotter's 8 Step Change Model

Implementing and Sustaining New Ways

Monitor uptake and governance ("Sustaining Change")

Enabling and Engaging the Organization

Prepare people to work in the new environment ("Training")

Demonstrate how the change will work ("Orientation")

Explain how the change will affect individuals ("Familiarization")

Setting the Climate for Change

Create an understanding of the rationale & opportunities ("Positioning Change")

8. Making it stick

7. Consolidating gains and building on success

6. Generating and celebrating short-term wins

5. Enabling employees to make the change

4. Communicating the change and mobilizing commitment

3. Developing a vision and strategy

2. Developing and being part of a guiding coalition

1. Creating a sense of shared need and urgency

Managing individual change with ADKAR®

- ADKAR® describes the key building blocks for successful change
 - Personal or professional
- Success with change requires all elements of the ADKAR® Model to be present

Reference: Hiatt, J. *ADKAR: A model for change in business, government and our community*, Learning Center Publications, 2006.

ADKAR is a registered trademark of Prosci. All Rights Reserved

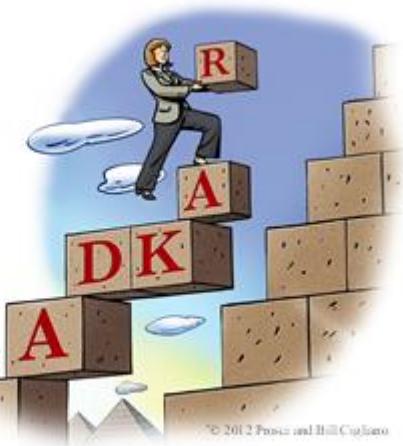
Awareness

Desire

Knowledge

Ability

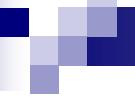
Reinforcement®



© 2012 Prosci and Bill Capilano

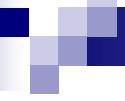
Awareness - factors influencing success

- A person's view of the current state
- How a person perceives problems
- Credibility of the sender of the awareness messages
- Circulations of misinformation/rumors
- Contestability of the reasons for change



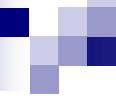
Desire – factors influencing success

- The nature of the change (what impact will it have on each person)
- The organizational and environmental context
- Each individual person's situation
- What motivates a person



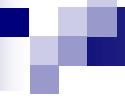
Knowledge

- The current knowledge base of the individual
- The capability of a person to gain additional knowledge
- Available Resources for training/education
- Access to or existence of the required knowledge



Ability

- Psychological blocks
- Physical capabilities
- Intellectual capability
- The time available to develop the needed skills
- Availability of resources to support the development of new abilities



Reinforcement

- The degree to which reinforcement is meaningful
- The association of reinforcement with actual demonstrated progress
- The absence of negative consequences
- An accountability system that enables the reinforcement of change

Prosci's organizational change management process

A structured process for managing the ‘people side’ of change on a project or initiative



Phase 1 – Preparing for change

- Understanding the nature of the change
- Understanding the groups being changed
- Creating the right sponsorship model and coalition
- Identifying risks
- Developing special tactics

Phase 1 - Preparing for change

Define your change management strategy

Prepare your change management team

Develop your sponsorship model

Phase 2 – Managing change

- Communication plan
- Sponsor roadmap
- Training plan
- Coaching plan
- Resistance management plan

Phase 2 - Managing change

Develop change
management plans

Take action and
implement plans

Phase 3 – Reinforcing change

- Compliance audit reports and employee feedback
- Corrective action plans
- After action review
- Transition management

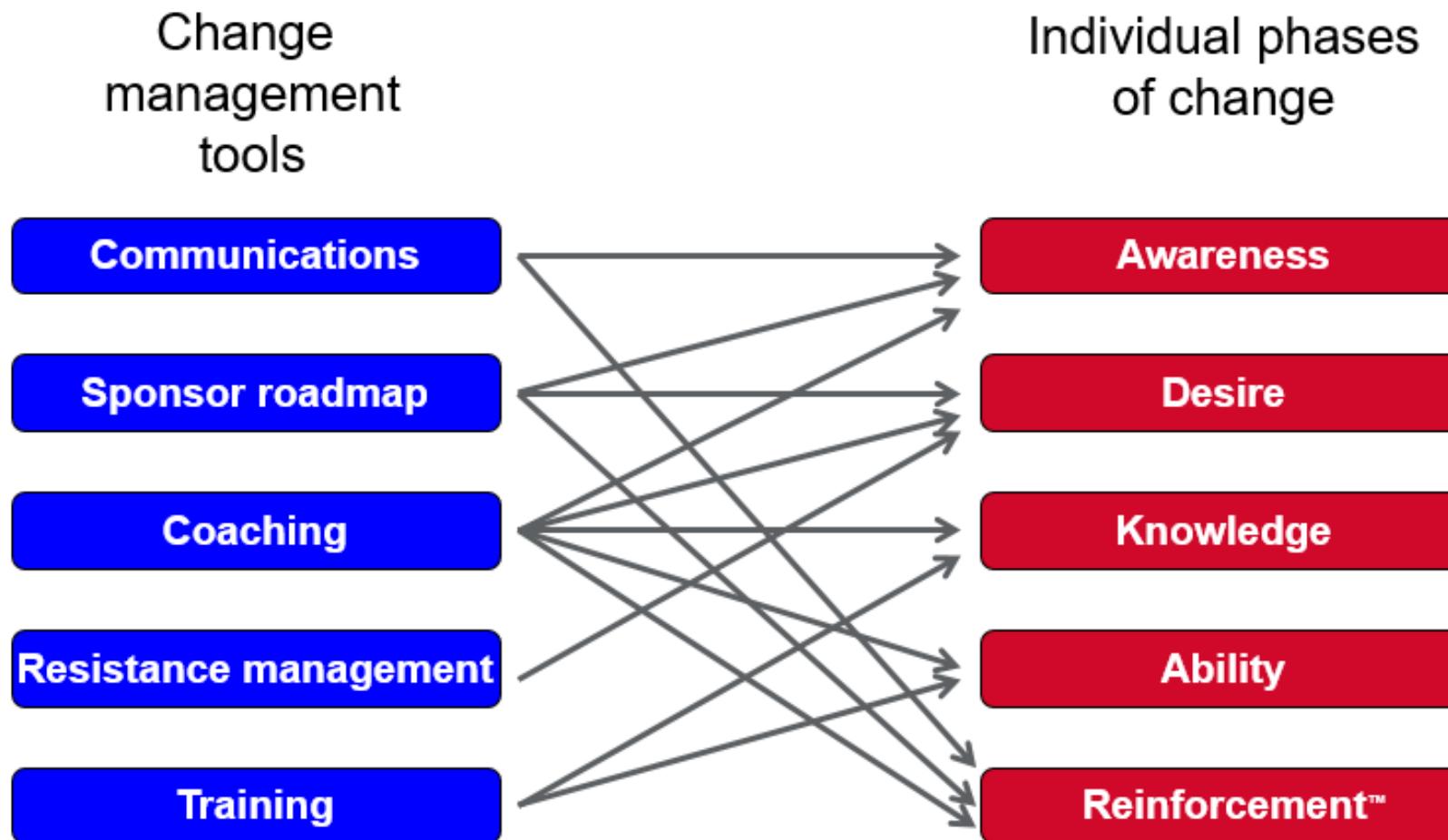
Phase 3 - Reinforcing change

Collect and analyze feedback

Diagnose gaps and manage resistance

Implement corrective actions and celebrate successes

Individual and organizational change management



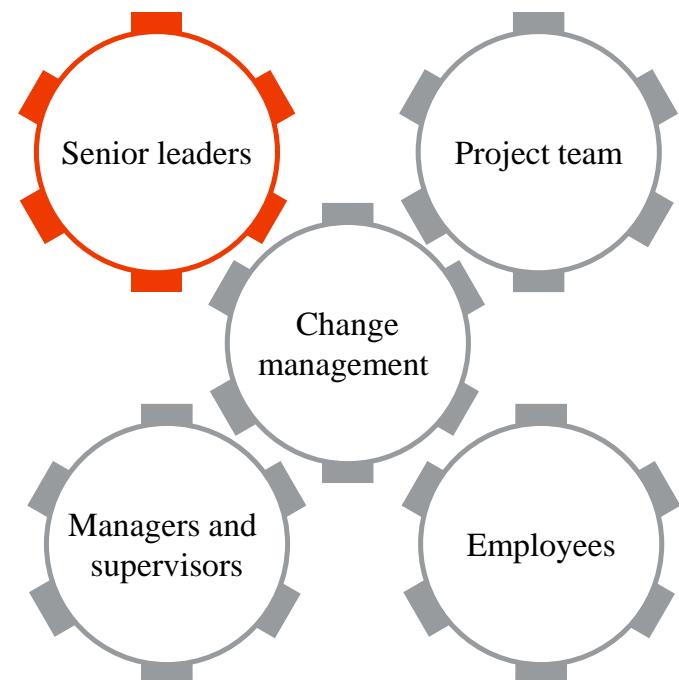
Who is involved in managing change

- The change management resource on a project plays the role of enabler
 - The conductor of the orchestra
 - The director of the play
- Effective change management requires involvement and action by many in the organization



Senior leaders

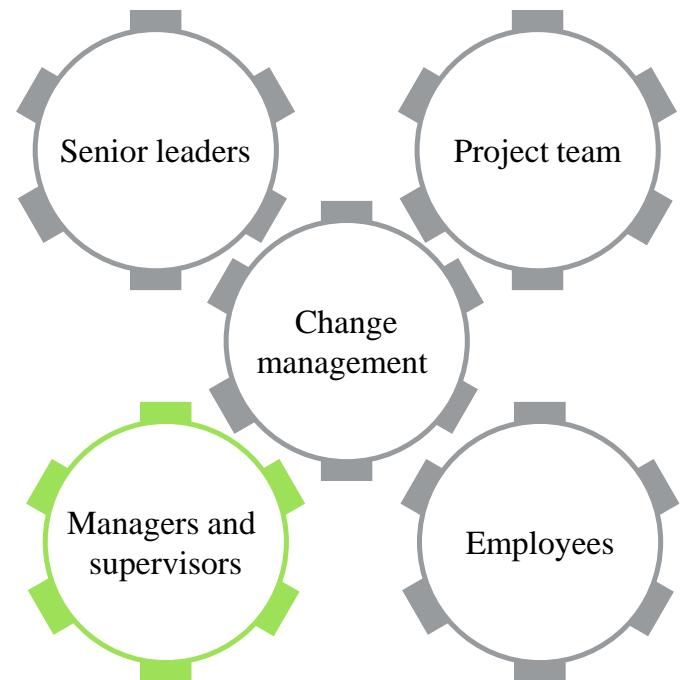
- Why is this group important?
 - Active and visible sponsorship is identified as the top contributor to overall project success in Prosci's five benchmarking studies
 - Senior leaders are one of two preferred senders of messages about change
- What is this group's role?
 - Participate actively and visibly throughout the project
 - Build the needed coalition of sponsorship with peers and other managers
 - Communicate the business messages about the change effectively with employees



Managers & supervisors

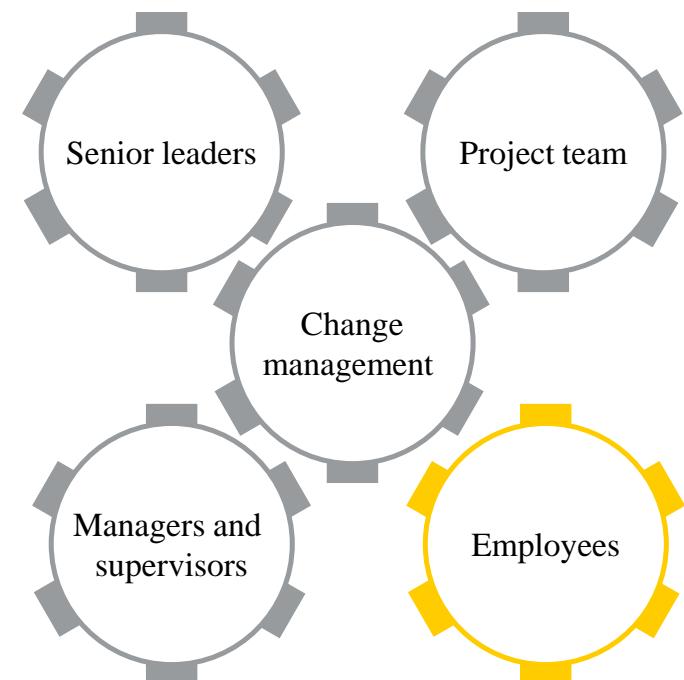
- Why is this group important?
 - Managers and supervisors are the other preferred sender of messages about change
 - This group has a unique and well-developed relationship with the employees being impacted by the change

- What is this group's role?
 - Communicate the personal messages about the change with their direct reports
 - Conduct group and individual coaching sessions
 - Identify, analyze and manage resistance
 - Provide feedback to the rest of the change management 'gears'



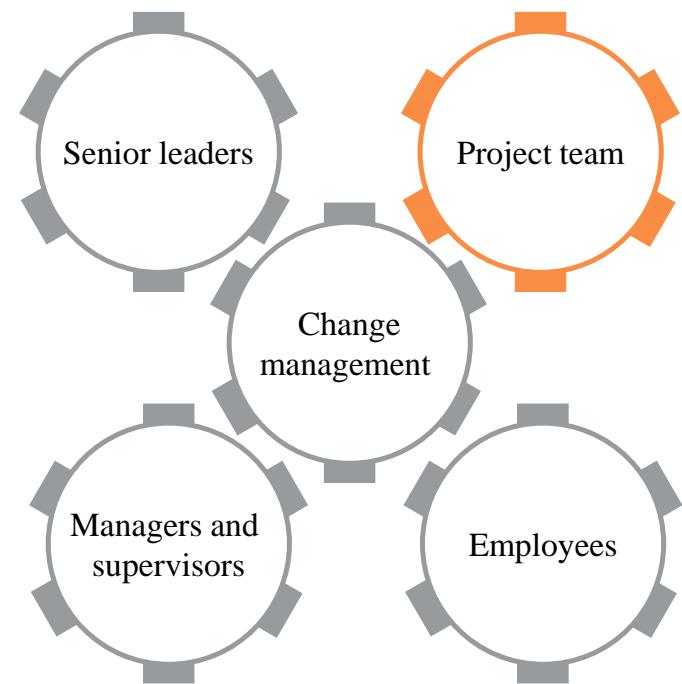
Employees

- Why is this group important?
 - Employees will ultimately make changes to how they do their day-to-day work
 - Their acceptance and use of the solution determines the success of the project and the ongoing benefit derived from the change
 -
- What is this group's role?
 - Seek out information related to the business reasons for change and the personal impact of the change
 - Provide feedback and reaction to the change and the change management efforts
 - Take control of the personal transition (using an individual change management model like ADKAR)

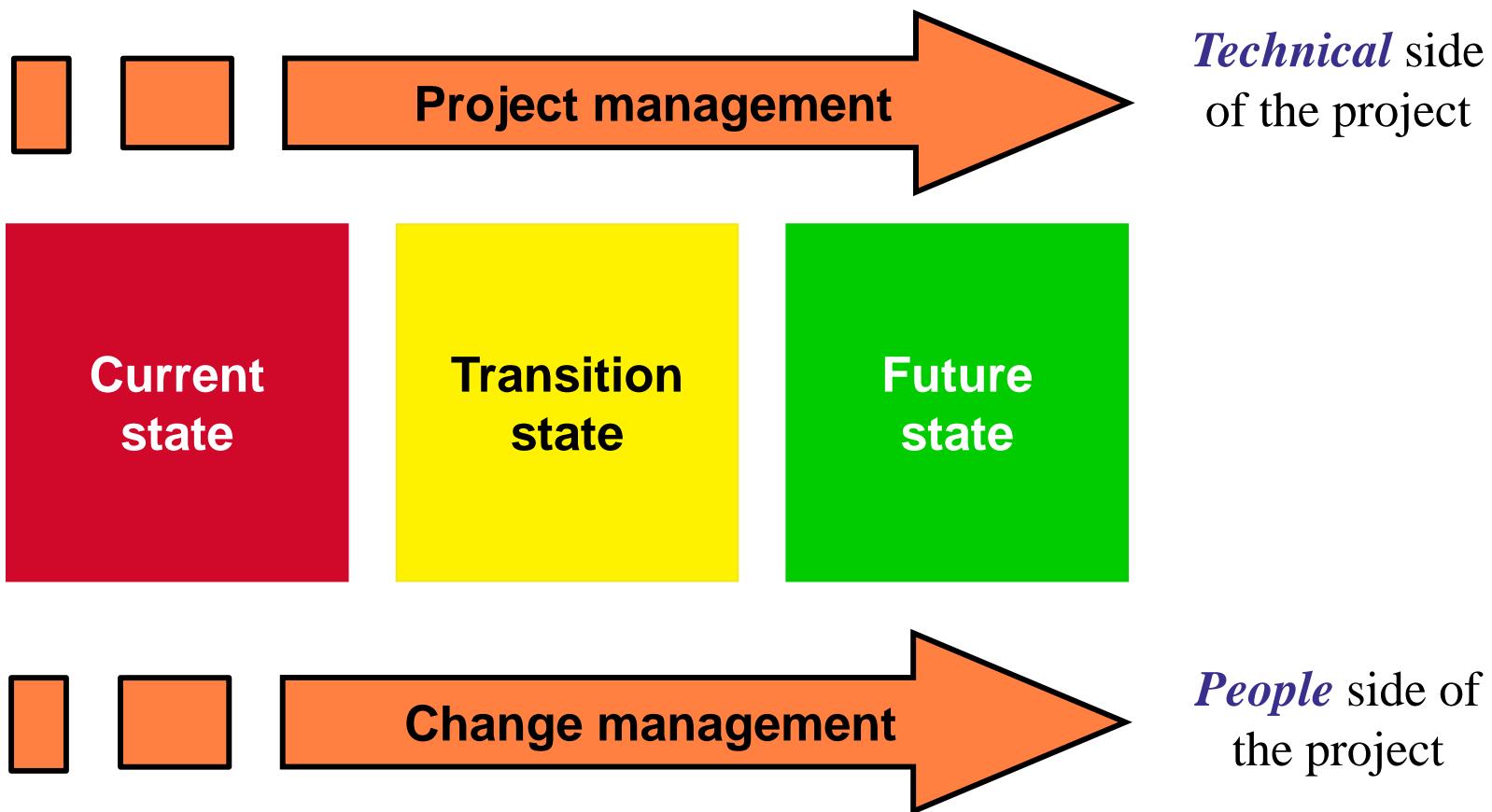


Project team

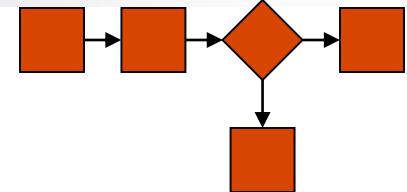
- Why is this group important?
 - The project team designs and develops the ‘change’ – they are the ones who introduce new processes, systems, tools, job roles and responsibilities
 - This group provides much of the specific information about the change to the other ‘gears’
- What is this group’s role?
 - Provide timely, accurate and succinct information about the change (or project)
 - Integrate change management activities into project management plans and activities



Change management and project management

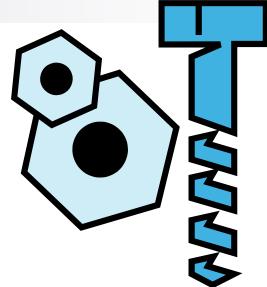


Comparing processes



- Project management
 - Initiation
 - Planning
 - Executing
 - Monitoring and controlling
 - Closing
- Process groups defined in the Project Management Institute's PMBOK®
- Change management
 - Organizational:
 - Preparing for change
 - Managing change
 - Reinforcing change™
 - Individual:
 - Awareness
 - Desire
 - Knowledge
 - Ability
 - Reinforcement®

Comparing tools



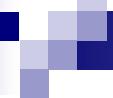
- Project management
 - Statement of work
 - Project charter
 - Business case
 - Work breakdown structure
 - Budget estimations
 - Resource allocation
 - Schedule
 - Tracking
- Change management
 - Individual change model
 - Readiness assessment
 - Communication plans
 - Sponsorship roadmaps
 - Coaching plans
 - Training plans
 - Resistance management
 - Reinforcement

The right amount

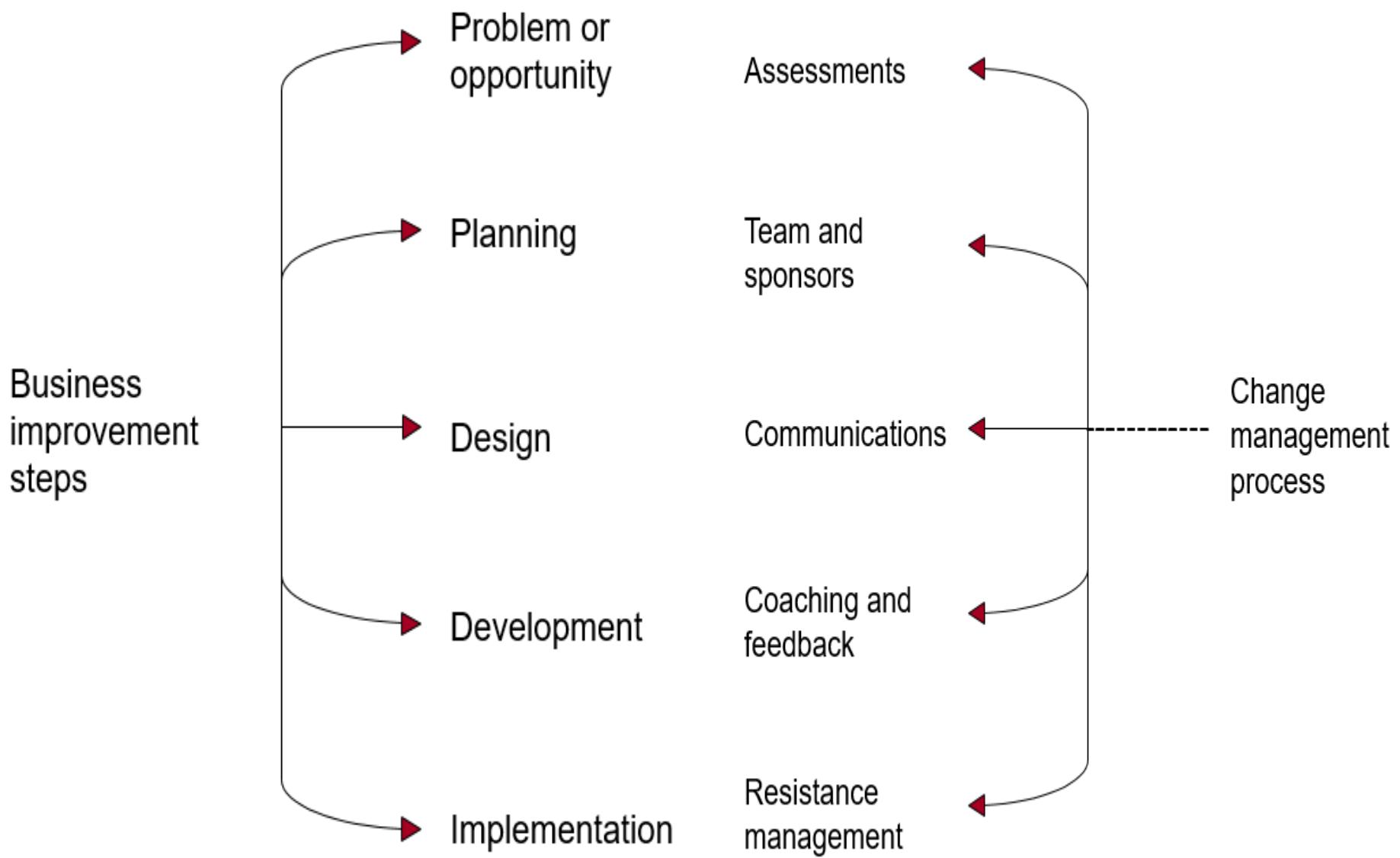
- How much *project management* is needed?
 - Depends on the complexity and degree of change to processes, systems, organization structure and job roles
- How much *change management* is needed?
 - Depends on the amount of disruption created in individual employees' day-to-day work and the organization attributes like culture, value system and history with past changes

Integrating change management and project management activities

- Project management and change management activities are most effective when they are integrated
- Unfortunately, in many instances change management is an add-on after the project has experienced obstacles



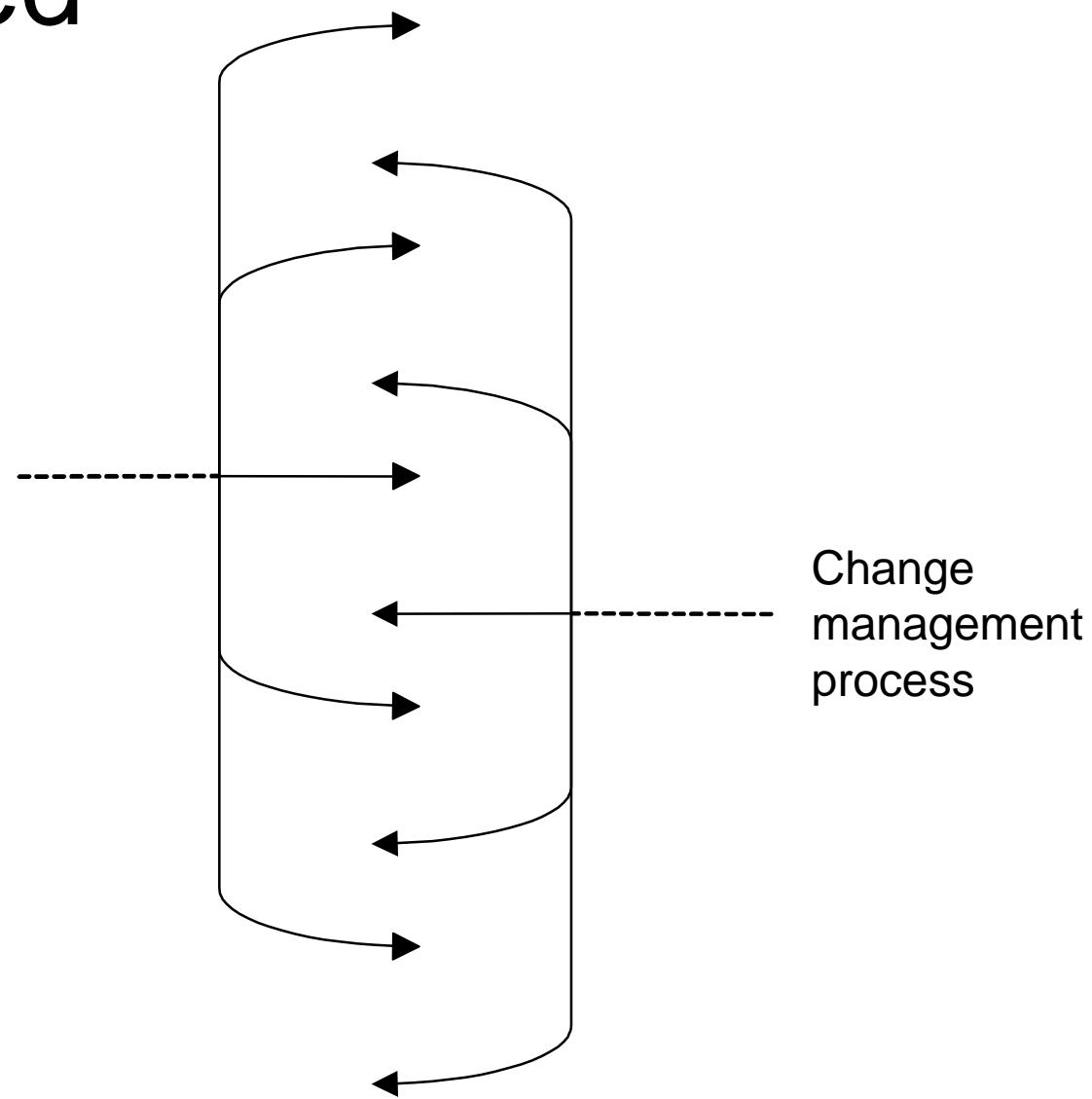
Business improvement steps



Integrated

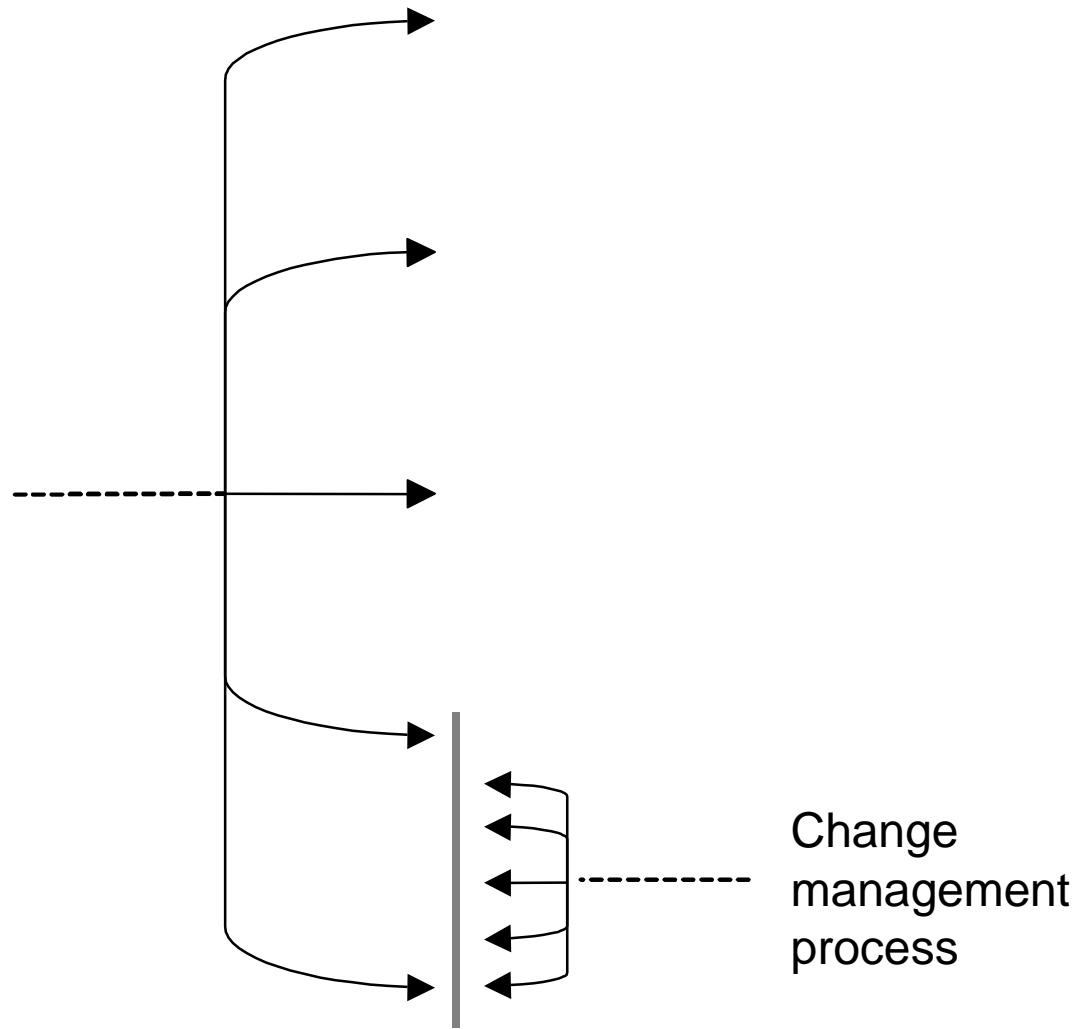
Business
improvement
steps

Change
management
process



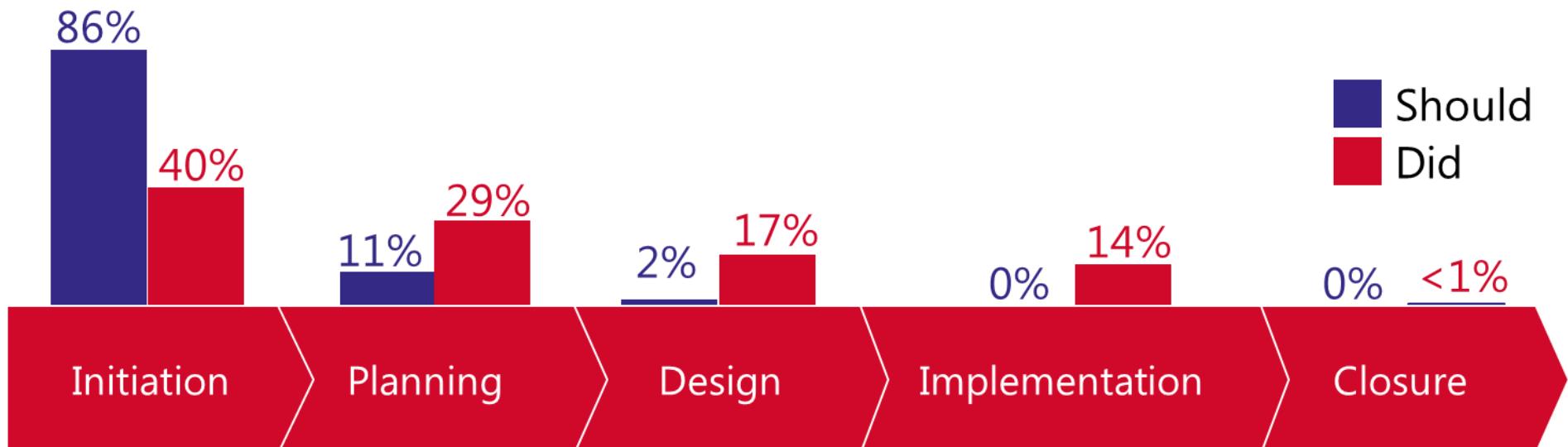
Too late

Business
improvement
steps



Research results

When to start change management



Final Conclusions

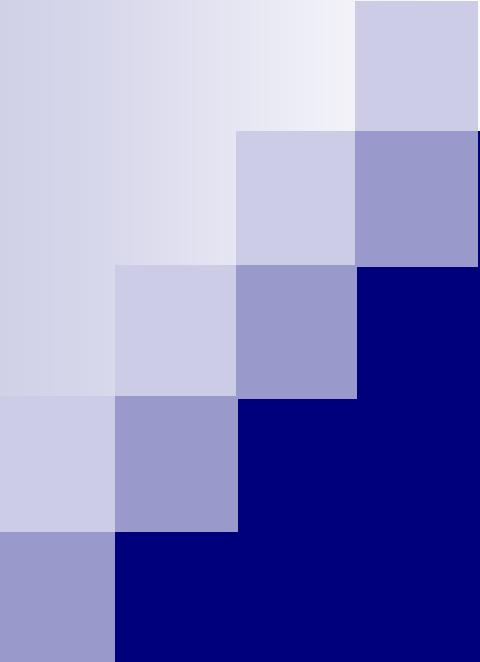
- What are CM objectives?
 - Managing change is more than just checking-in and checking-out files. It includes management of workspaces, parallel development, integration, and builds.
- Which are the main topics for CM?
 - Coordinating the Activities and Artifacts
 - Coordinating Iterations and Releases
 - Coordinating Changes to Software

Final Conclusions

- When CM is established?
 - Often institutionalized or established early on in the project lifecycle
 - CRs, can be raised at any time during the course of the project
- Who is responsible for CM?
 - Change Control Manager/Configuration Manager
 - Anyone on the project staff should be able to raise a Change Request

Final Conclusions

- Who tells us how CM can be done?
 - Several CM models

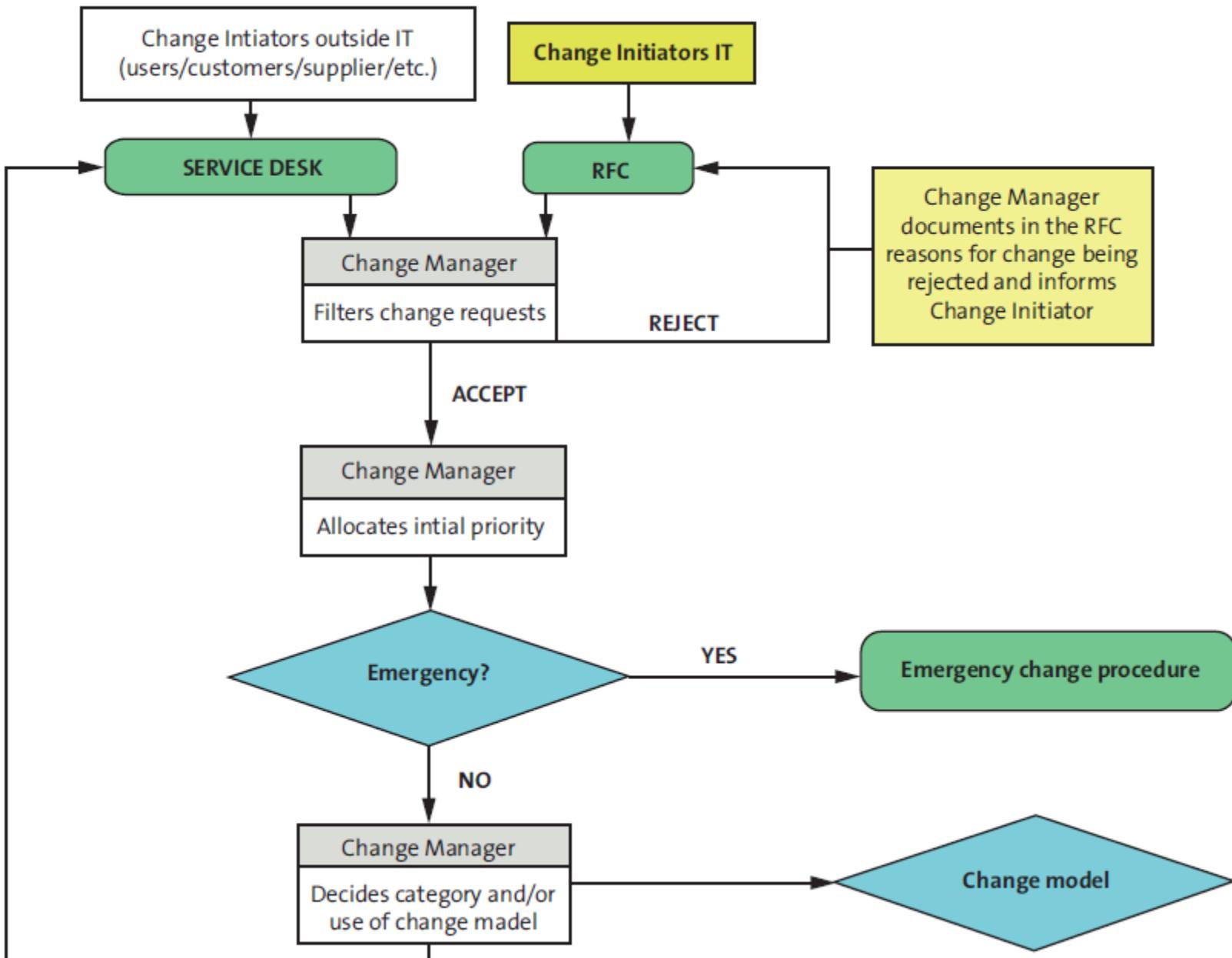


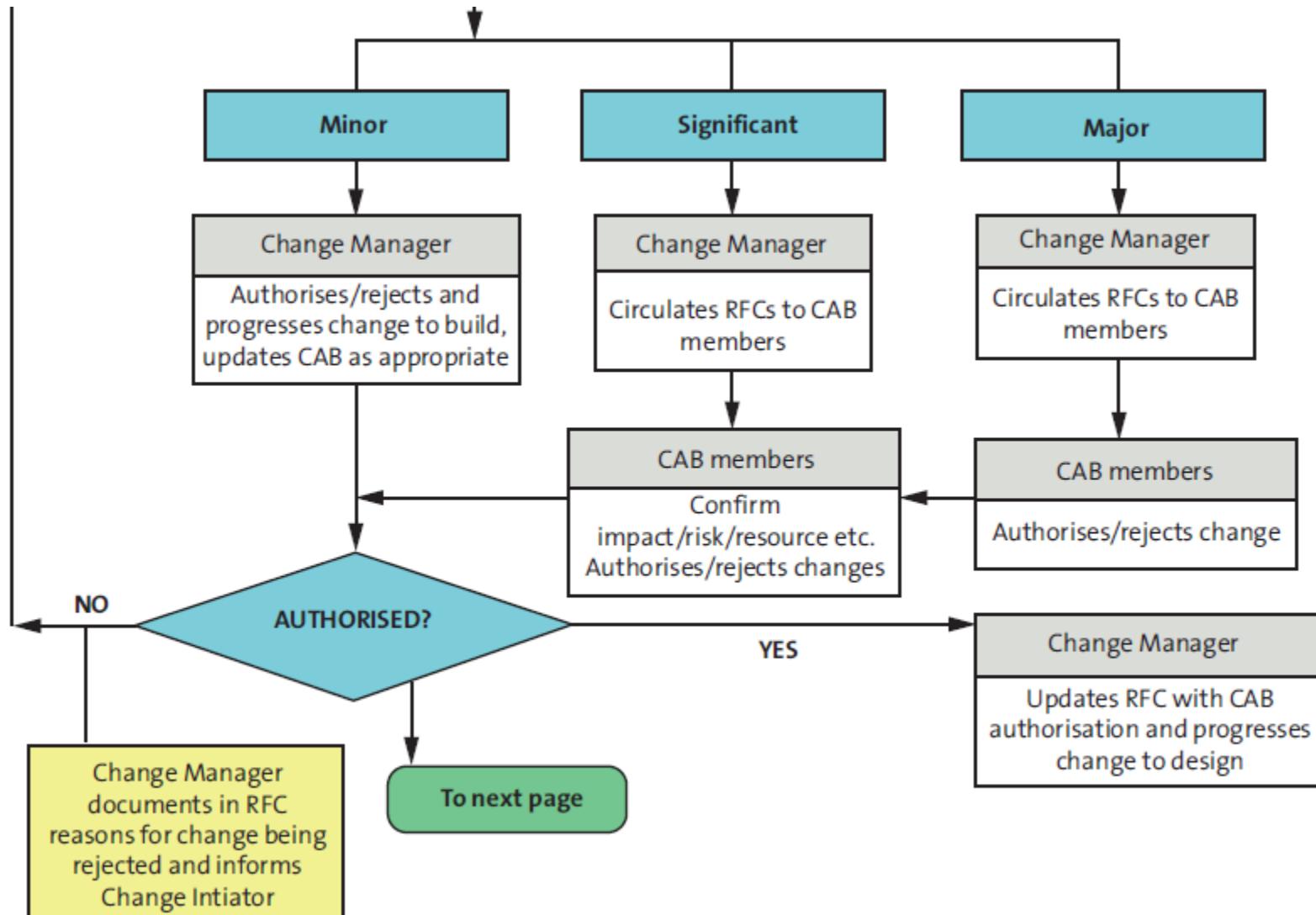
Project Closure

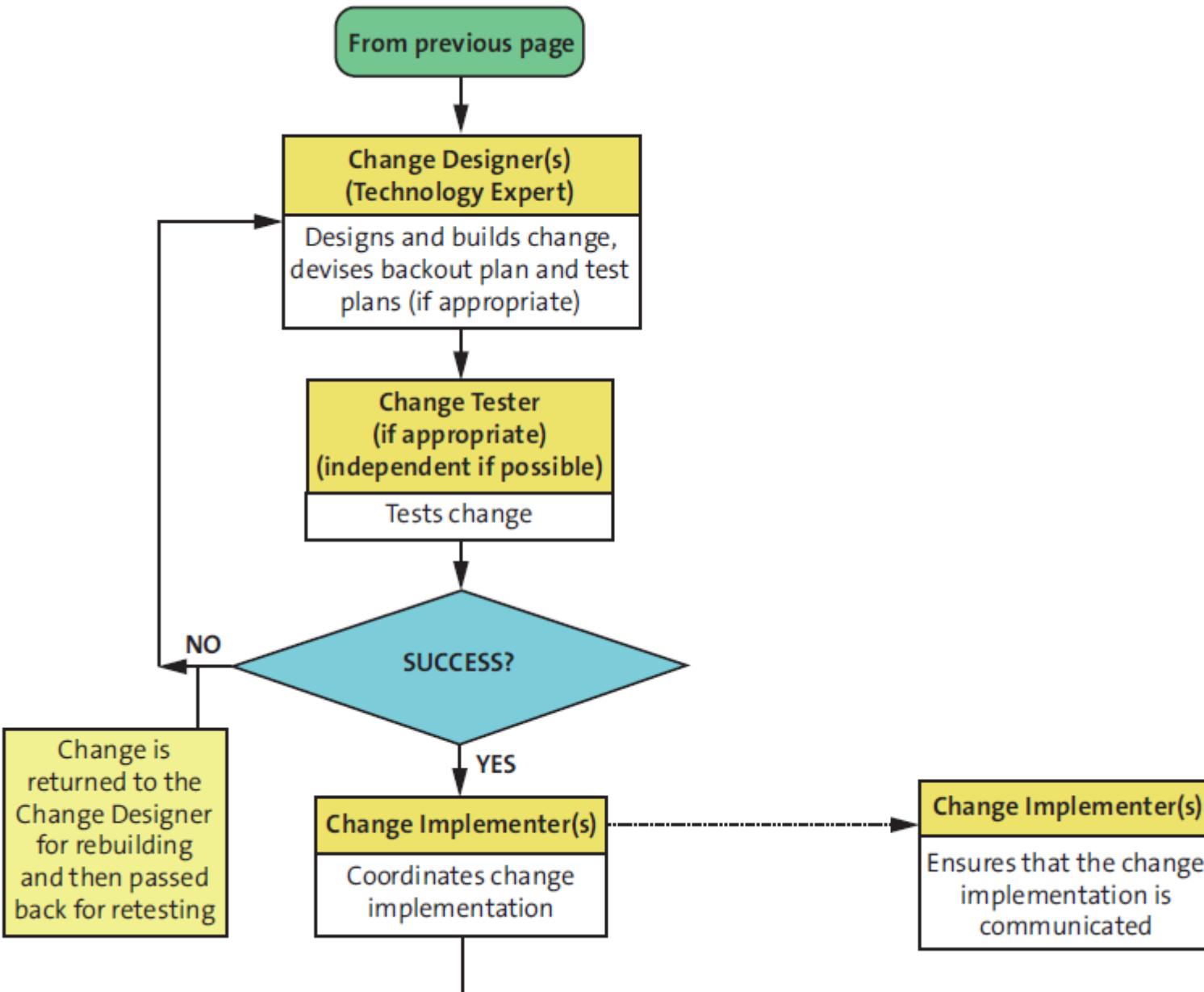
Lecture 12

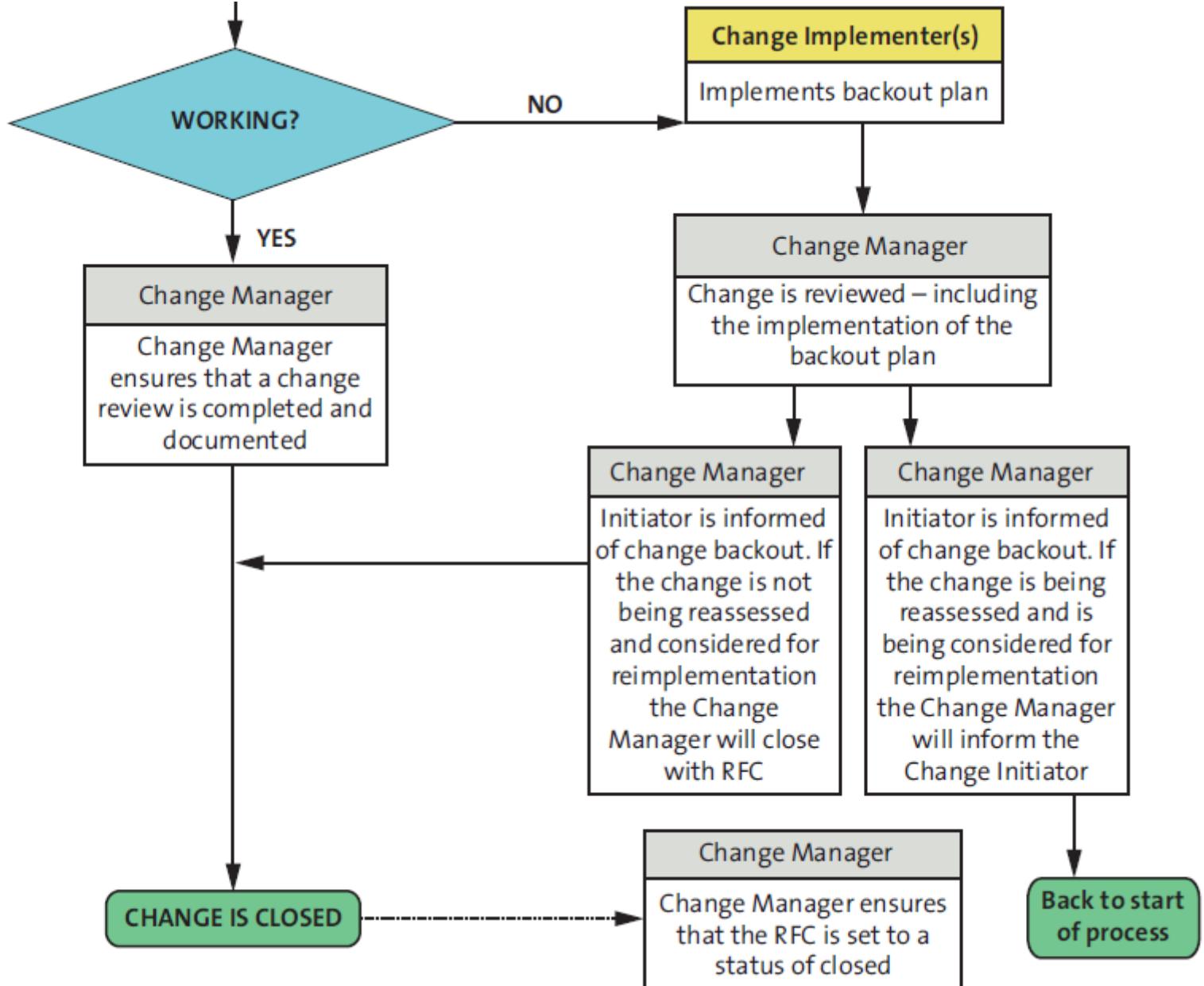
Recap: Change Management

- *ICC is the process necessary for **controlling** factors that create changes to make sure those changes are beneficial, **determining** whether a change has occurred, and **managing** the approved changes, including when they occur.*
- ICC process is performed throughout the project, from project initiation through project closure.

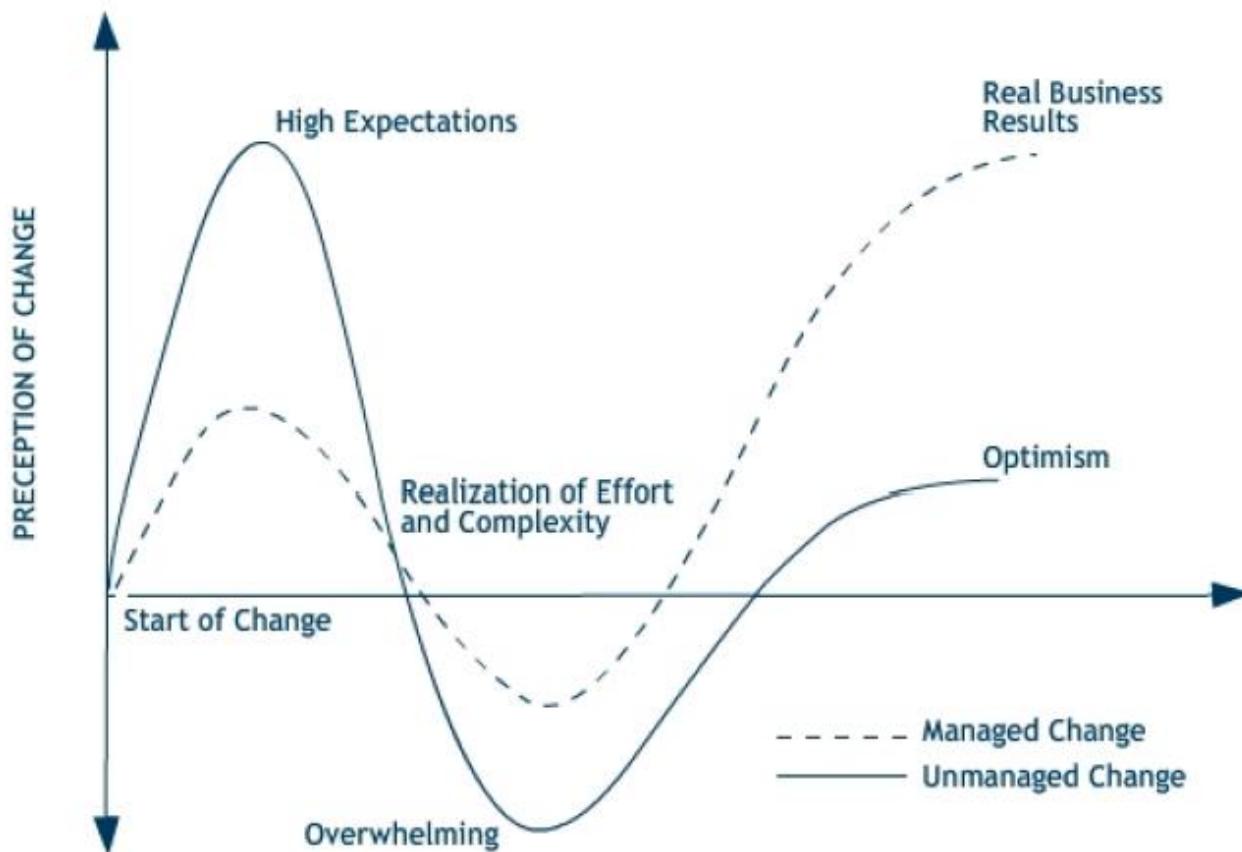








Recap:: Project change perception



Recap:: Kotter's 8 Step Change Model

Implementing and Sustaining New Ways

Monitor uptake and governance ("Sustaining Change")

Enabling and Engaging the Organization

Prepare people to work in the new environment ("Training")

Demonstrate how the change will work ("Orientation")

Explain how the change will affect individuals ("Familiarization")

Setting the Climate for Change

Create an understanding of the rationale & opportunities ("Positioning Change")

8. Making it stick

7. Consolidating gains and building on success

6. Generating and celebrating short-term wins

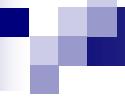
5. Enabling employees to make the change

4. Communicating the change and mobilizing commitment

3. Developing a vision and strategy

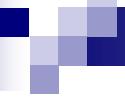
2. Developing and being part of a guiding coalition

1. Creating a sense of shared need and urgency



Today's topic

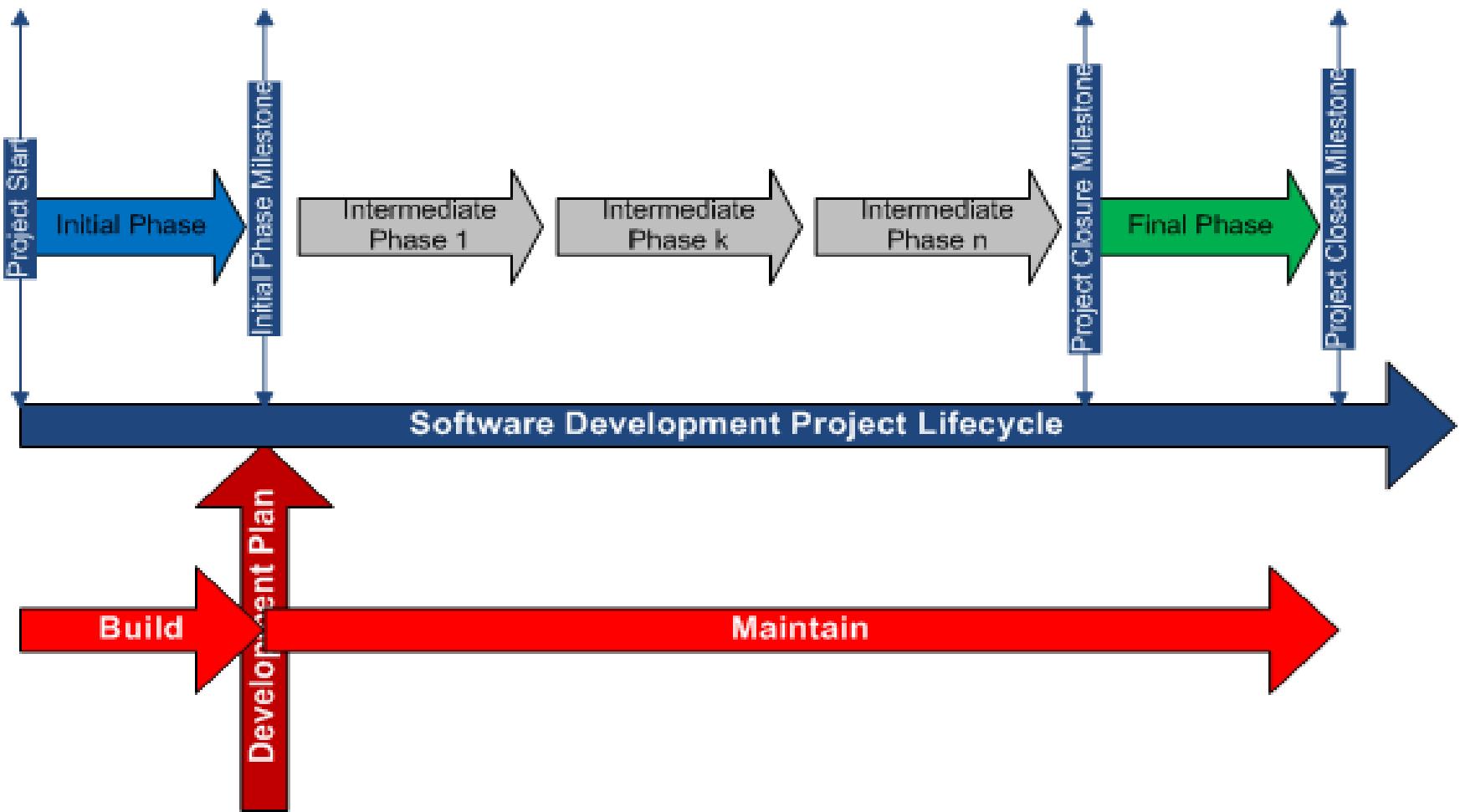
- Project Closure

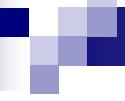


References

- PMBOK
- Managing Iterative Software Development Projects by Kurt Bittner, Ian Spence
- Project management, the managerial process by Erik Larson and Clifford Gray
- SCRUM Guide

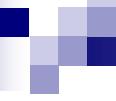
Project Lifecycle





Why Project Closure?

- Projects are **temporary** endeavours
- Without a **formal closure process**, project teams can fail to recognize the end, and then the project can drag on—sometimes at great expense
- Closure must be done both on project completion and cancelation



What does project closure?

- Ensures that
 - Outcomes match the stated goals of the project (when completion is achieved)
 - Customers and stakeholders are happy with the results
 - The team feels a sense of completion
 - Critical knowledge is captured
 - Project resources are released for new projects

Team

- One of the highest benefits from project closure is releasing the team
 - Formal
 - Post mortem

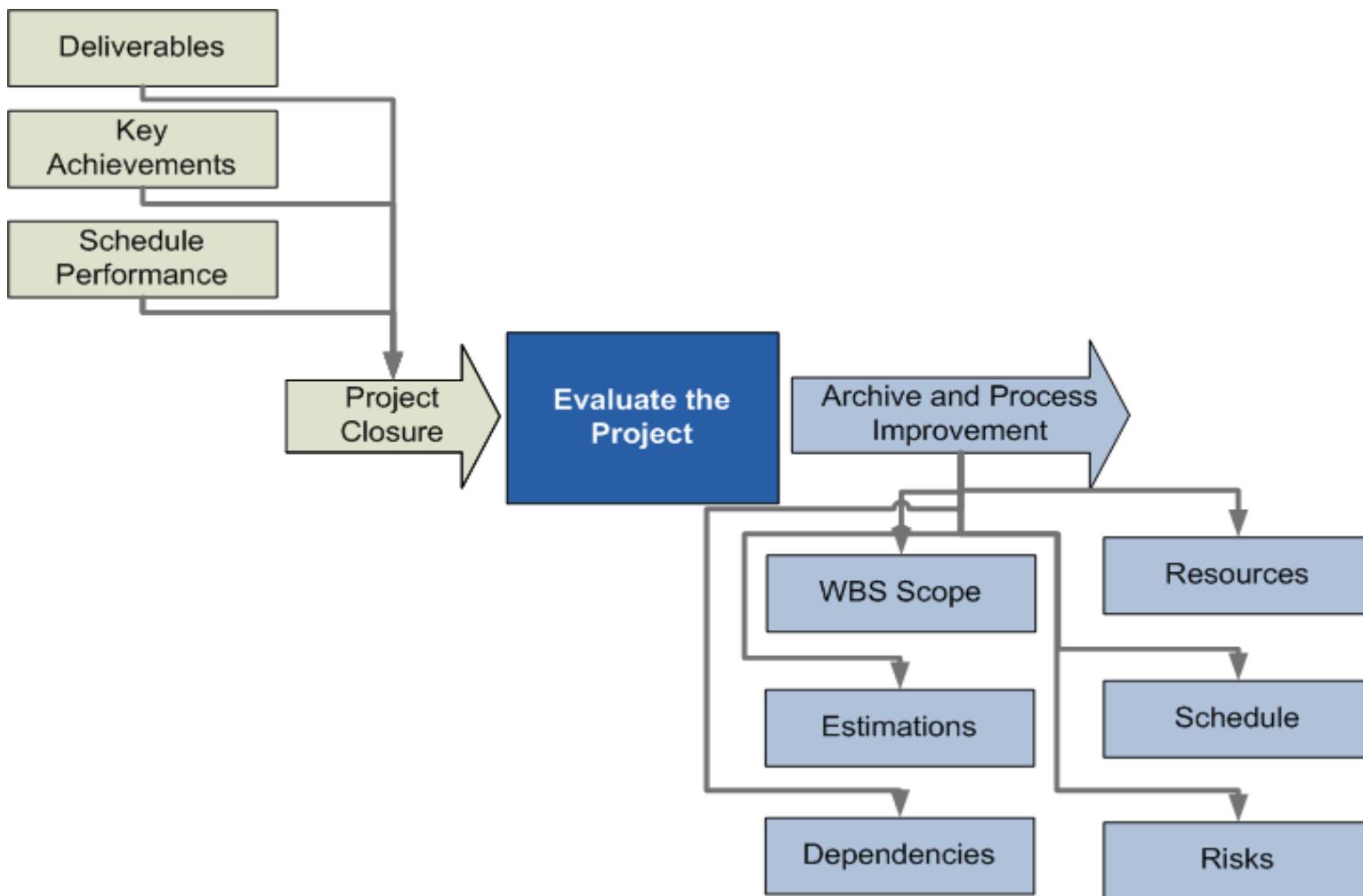
Types of project Closure

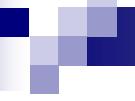
- Normal (objectives achieved)
- Premature (objectives achieved partially, some activities skipped)
- Perpetual (continuous add-ons)
- Failed project (objectives not achieved)
- Changed priority (objectives can be achieved, not relevant anymore)

The Project Closure Phase

- The last phase of the Project Life Cycle
- Determined by
 - the completion of all Project Objectives and acceptance of the end product by the customer OR
 - decision of closure
- The outputs from Project Closure Phase is input to execute the next projects with much more efficiency and control

Project Closure



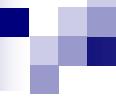


PMBOK – Closing Process Group

- Includes the processes to
 - Formally terminate all activities of a project or project phase
 - Hand over the project to maintenance teams
- The project can be either completed or canceled

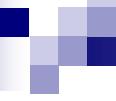
Closing Process Group

- Contains two processes:
 - **Project Closure:** finalize all activities across all of the Process Groups to formally close the project or a project phase.
 - **Contract Closure:** complete and settle each contract, including the resolution of any open items, and closing each contract applicable to the project or a project phase.



Project Closure Procedures

- includes:
 - Product Verification - Work completed correctly and satisfactory
 - Administrative Closure –
 - Integrated activities to collect project records
 - Analyze project success or failure
 - Gather lessons learned
 - Archive project for future use
- Input for the Contract Closure Process



Project Closure Inputs

- Project Management Plan
- Contract documentation
- Enterprise environmental factors
- Organizational process assets
- Work performance information
- Deliverables

Inputs: Contract Documentation

- Used to perform the contract closure process
- Includes
 - The contract itself
 - All changes to the contract
 - Other documentation (such as the technical approach, product description, or deliverable acceptance criteria and procedures).

Inputs: Enterprise environmental factors

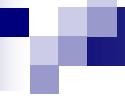
- Organizational or company culture and structure
- Governmental or industry standards
- Infrastructure
- Existing human resources
- Personnel administration
- Company work authorization system
- Marketplace info, stakeholder risk tolerance...

Inputs: Organizational process assets

- Organization's processes and procedures for conducting work
 - Standardized guidelines, work instructions, proposal evaluation criteria, and performance measurement criteria, templates...
- Organizational corporate knowledge base for storing and retrieving information
 - Process measurement database, project files,...

Inputs: Work performance information

- Schedule progress
- Deliverables that have been completed and those not completed
- Schedule activities that have started and those that have been finished
- Extent to which quality standards are being met
- Costs authorized and incurred
- ...



Project Closure Outputs

- Administrative Closure Procedure
- Contract Closure Procedure
- Final Product, service or result
- Organizational process assets (updates)

Outputs: Administrative Closure Procedure

- All procedures and roles involved in administrative closure
- Procedures for transfer to production
 - Action and activities for stakeholder approval
 - Confirmation of goals achievement (sponsor, client, stakeholders, etc)
 - Actions for satisfying all exit criteria

Outputs: Contract Closure Procedure

- A step-by-step methodology addressing:
 - Terms and conditions of the contracts and any required completion or exit criteria for contract closure
 - Activities and responsibilities of the project ALL team members involved in the contract closure process
- Actions performed to formally close all contracts associated with the completed project

Output: Final Product, Service or Result

- Formal acceptance and handover of the final product, service, or result that the project was authorized to produce
- The acceptance includes receipt of a formal statement that the terms of the contract have been met

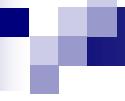


Output: Organizational Process Assets

- Formal Acceptance Documentation
- Project Files
- Project Closure Documents
- Historical Information

Close Process Tools

- Project Management Methodology
- Project Management Information System
- Expert Judgment



Before Tailoring Project Closure

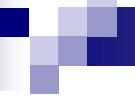
- Understand Iteration Closure
- Phase Closure
- ... and then Project Closure

Iteration Closure

- Iteration is concluded by the iteration acceptance review, where the formal result of the iteration is agreed upon and recorded
- This involves considering the results of the iteration and analyzing their impact on the project as a whole

Iteration Closure: Measurement and Analysis

- Absolute progress
- Risk exposure
- Estimate to complete
- Effort profile
- Cost profile
- Effort Expended
- Find/fix rate (tolerance)
- Defect trends (tolerance)



Iteration Closure: Acceptance Review

- Results
- Risks
- Objectives
- Requirements
- Adherence to Plan
- Lessons Learned
- Rework, external changes, feedback from demo, other deliverables...

Iteration Grades

- Exceptional
- Passed
- Passed at risk
- Unfinished
- Failed
- Abandoned

Iteration recommendations

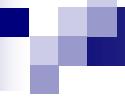
- For *Unfinished, Failed or Abandoned* we can have the following recommendations:
 - Re-plan
 - De-scope
 - Extend
 - Try again
 - Cancel

Acting on Iteration assessment results

- Never confuse the map with the journey
- Adopt an attitude that continuous planning is good
- Mature your process alongside your team
- Be prepared to cut your losses
- Be honest

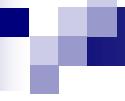
Phase Assessment

- Differences from iteration assessments:
 - Judge the project, not a single iteration
 - Always decide whether to continue or cancel the project
 - Focus on the delivered value rather than on performance
 - Assess project against the business case



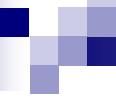
Phase Assessment topics

- Progress
- Risk
- Scope
- Baseline
- Performance
- Plans
- Business Case



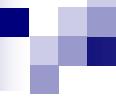
Phase Assessment approaches

- Formal Phase Assessment
- Extend Iteration Assessment



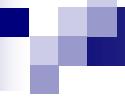
Concluding a Phase

- Passed
- Passed at risk
- Failed



Project conclusions

- Approved
- Temporary approved
- Extended
- Paused
- Canceled

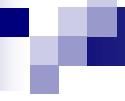


Project Assessments

- Performed not only at project closure
- Cover for:
 - Handle exceptions raised by the iteration assessments
 - Phase assessments are too far apart
 - Provide review points for the overall project (might include non-software subprojects)

Assessment objectives [1]

- Confirm that the need for the project has not changed
- Satisfy the stakeholders of the quality of the products delivered by the project
- Confirm that the business case is still viable



Assessment objectives [2]

- Authorize the continuation or cancellation of the project
- Assess the benefits that the project has delivered
- Assess the overall effectiveness of the project

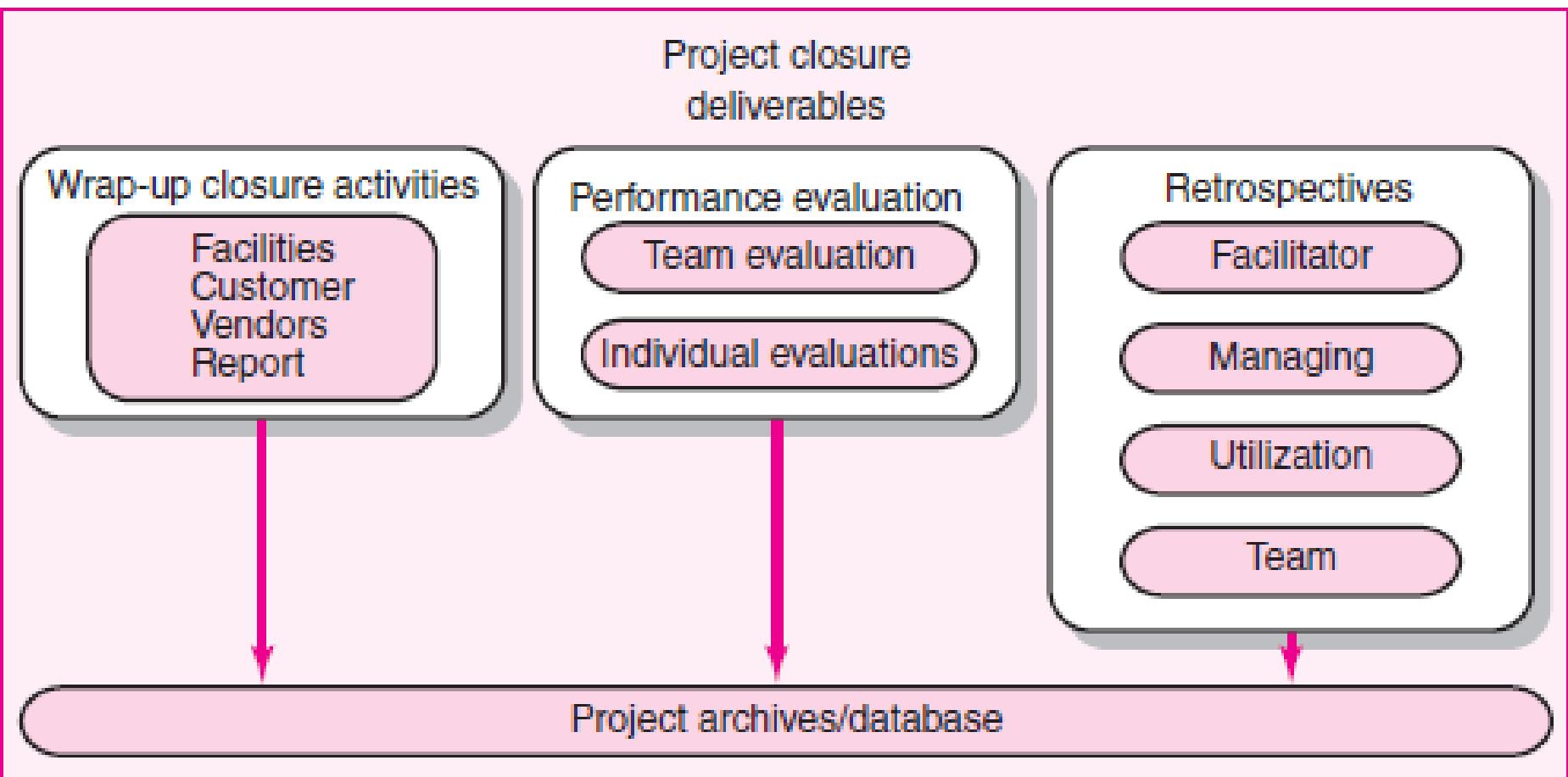
Checklist

1. Has the product been implemented and is it producing the expected benefits?
2. Have all deliverables been completed and implemented? If not, has accountability for outstanding items been assigned?
3. Have new workflows and procedures been documented, tested and implemented?
4. Have all outstanding issues been resolved or assigned owners?
5. Have hardware, software and other components been installed?

Checklist continued

6. Are system, functional, performance and user acceptance testing complete? Have serious bugs been fixed and a process implemented to deal with smaller bugs and/or new bugs that might surface during regular operations?
7. Is training of all staff (users, support, help desk) complete?
8. Is all required system and user documentation complete and available?
9. Has any old hardware, software, licenses been decommissioned?
10. Do you have the formal approval of the project from the Sponsor?

Deliverables



Wrap-up closure activities

- Getting delivery acceptance from the customer.
- Shutting down resources and releasing to new uses.
- Reassigning project team members.
- Closing accounts and seeing all bills are paid.
- Delivering the project to the customer.
- Creating a final report.

Example checklist

	Task	Completed? Yes/No
	<i>Team</i>	
1	Has a schedule for reducing project staff been developed and accepted?	
2	Has staff been released or notified of new assignments?	
3	Have performance reviews for team members been conducted?	
4	Has staff been offered outplacement services and career counseling activities?	
	<i>Vendors/contractors</i>	
5	Have performance reviews for all vendors been conducted?	
6	Have project accounts been finalized and all billing closed?	
	<i>Customer/Users</i>	
7	Has the customer signed-off on the delivered product?	
8	Has an in-depth project review and evaluation interview with the customer been conducted?	
9	Have the users been interviewed to assess their satisfaction with the deliverables? With the project team? With vendors? With training? With support? With maintenance?	
	<i>Equipment and facilities</i>	
10	Have project resources been transferred to other projects?	
11	Have rental or lease equipment agreements been closed out?	
12	Has the date for the closure review been set and stakeholders notified?	
	<i>Attach comments or links on any tasks you feel need explanation.</i>	

Final report

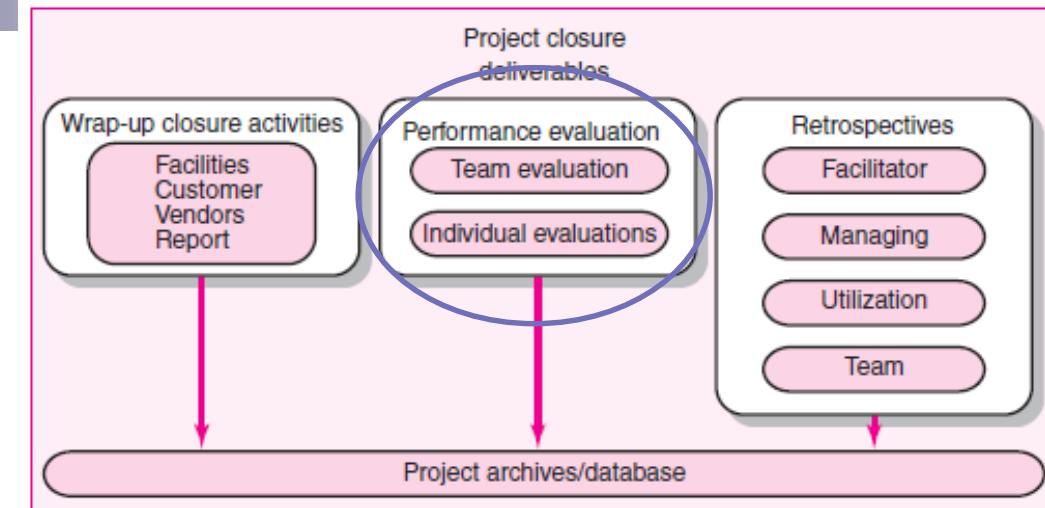
- Executive summary
- Review and Analysis
 - factual review statements of the project
 - underlying causes of problems, issues, and successes
- Recommendations
- Lessons learned

Performance evaluation

■ Team evaluation

□ Preconditions:

- Do standards for measuring performance exist? Are the goals clear for the team and individuals? Challenging? Attainable? Lead to positive consequences?
- Are individual and team responsibilities and performance standards known by all team members?
- Are team rewards adequate?
- Is the team empowered to manage short-term difficulties?
- Is there a relatively high level of trust emanating from the organizational culture?



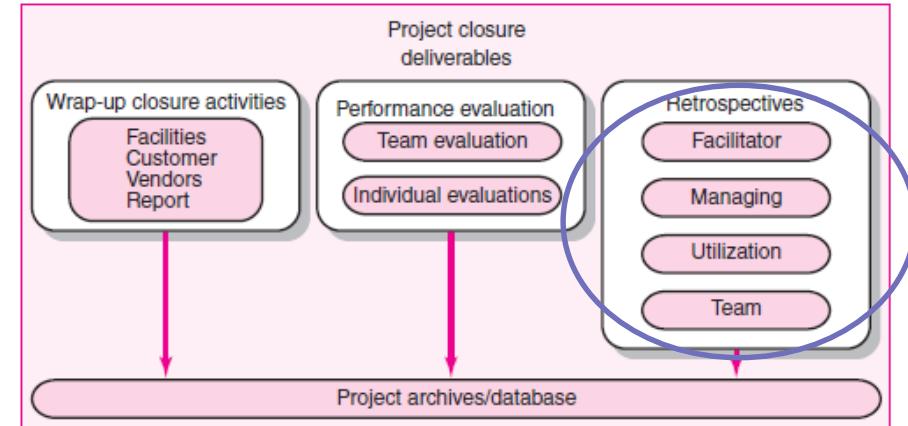
Team evaluation

- Team evaluation should go beyond time, cost, and specifications.
- Evaluation of the team-building process.
- Effectiveness of group decision and problem-solving processes, group cohesion, trust among team members, and quality of information exchanged.
- Measurement of customer and user satisfaction with project deliverables

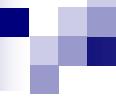
Individual evaluation

- 2 objectives:
 - Identify strengths and weaknesses => action plans for improving performance
 - Assess performance => salary/promotion/merits
- “360-degree feedback”
 - Feedback from project and area managers, peers, subordinates, and even customers.

Retrospectives

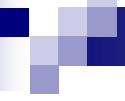


- *a methodology that analyzes a past project event to **determine what worked and what didn't**, develops **lessons learned**, and creates an **action plan** that ensures lessons learned are used to improve management of future projects.*



Retrospectives best practices

- Uses an independent facilitator.
- Includes a minimum of three in-process learning gates during the project life cycle.
- Has an owner.
- Develops a repository that is easy to use.
- Mandates a discipline that ensures retrospectives are used.



Project Closure in Agile

- Daily assessment
- Sprint assessment

Daily assessment

- Sprint Burn Down
 - work remaining to complete the team's sprint backlog, compared to the team capacity
- Cumulative Flow Diagram
 - flow of the sprint's user stories through the different statuses (total, new, in progress, in testing, done)
- Team Velocity
 - number of story points completed by the team in the current sprint to-date, compared to the team's average sprint output.
- Team Workload
 - work remaining or individual team members, compared to the member's remaining capacity.

Sprint review

- Demo
 - the stories that were developed
 - the story points that were covered
 - the new average velocity
 - other relevant info (changes, support material)
- Retrospective
 - What went well/wrong
 - Action Plan or Action Items
- Backlog Grooming

Sprint closing

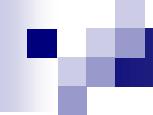
- At the end of each sprint, all open work is either:
 - Closed (if done)
 - Split (if partially done)
 - Moved forward to another sprint (not done)
 - or placed back into the product backlog (not done)

Conclusion

■ Formal project closure ensures that your team has:

- met its objectives,
- satisfied the customer,
- captured important knowledge,
- and been rewarded for their efforts.

With the door closed securely behind you, you can move on to your next project with Confidence.



"Let our advance worrying become advance thinking and planning."

Winston Churchill

Thank you and Good luck!

Final exam

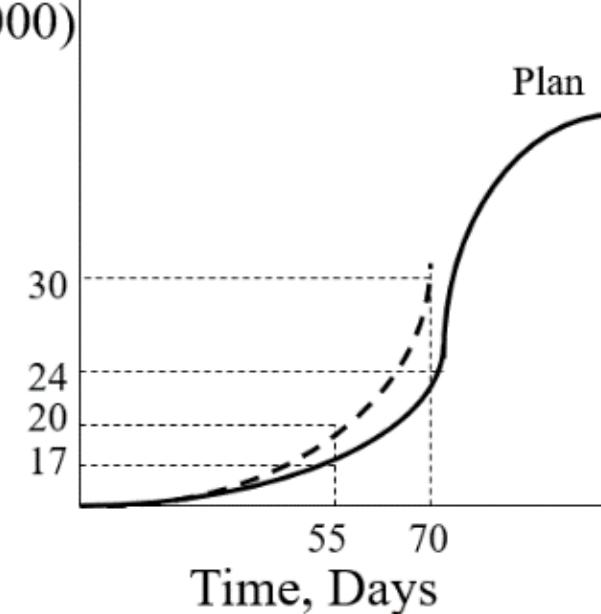
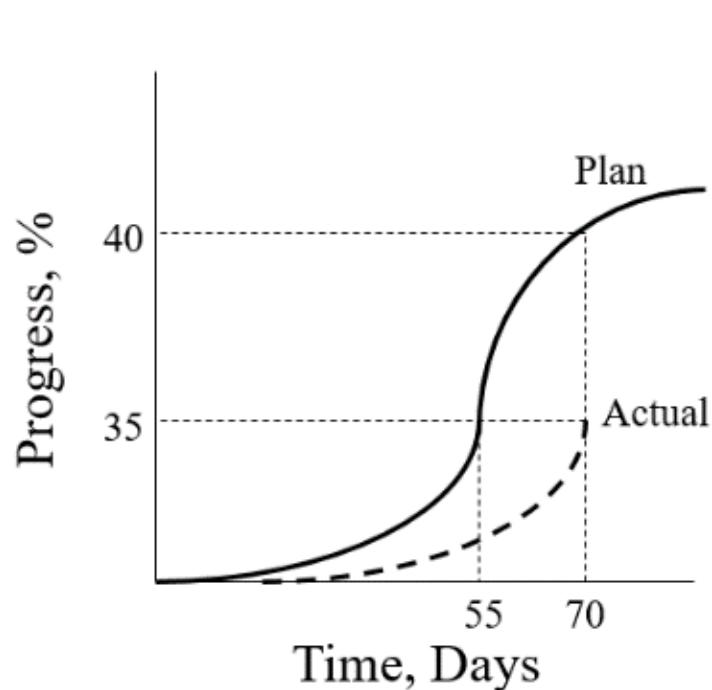
- Final exam according to schedule
- Online, using Moodle
- Timed quiz (multiple choice questions + essay questions)

Sample exam problems

Consider the following activities table. Construct a network diagram based on the activity descriptions. For the constructed network diagram: Identify the Critical Path(s). If you would need to reduce the length of the project by 2 weeks what approach would you take?

ACT	DUR	PREDECESSORS
B	5	
M	4	B
N	9	B
Q	15	B
A	1	M,N
F	4	N,Q
X	9	Q
C	9	Q
Y	9	A,F,X
S	6	F
J	5	X,F
T	10	C
V	5	Y,S
U	10	V,T,J

- Considering the following diagrams, what is the status of the project in day 70. Assuming that the variances are typical for the remaining of the project, what would be an Estimate at completion?



- You start a project and one of the significant requirements is very unclear. The customer promises to deliver a detailed specification soon. You identify a risk here. What response options are you considering (give at least 3 specific strategies and the category they belong to)?