

A central green circle contains the text "Project Initiation" in large, bold, blue letters. Surrounding this central circle are several concentric green rings. Radiating from the center are ten lines, each ending in a small circle. These outer circles are colored in a sequence of green, yellow, and blue. The background is a light gray with a subtle pattern of small dots.

Project Initiation

Copyright © 2001- 2017 NUS. The contents of this document may not be reproduced in any form or by any means, without the written permission of ISS, other than for the purpose for which it has been supplied

Total Number of slides = 66

TABLE OF CONTENTS

MODULE OBJECTIVES

01

Initiation
Process

02

Stakeholder
Analysis

03

Management
Strategy

AT THE END OF THIS MODULE...

You would need to be able to :

Define - The key activities in initiation

Identify and Analyse - Project Stakeholders

Define – A Management strategy

Initiation Process

-
- + *Definition*
 - + *Stakeholder Analysis*
 - + *Charter*
 - + *Kickoff meeting*

01

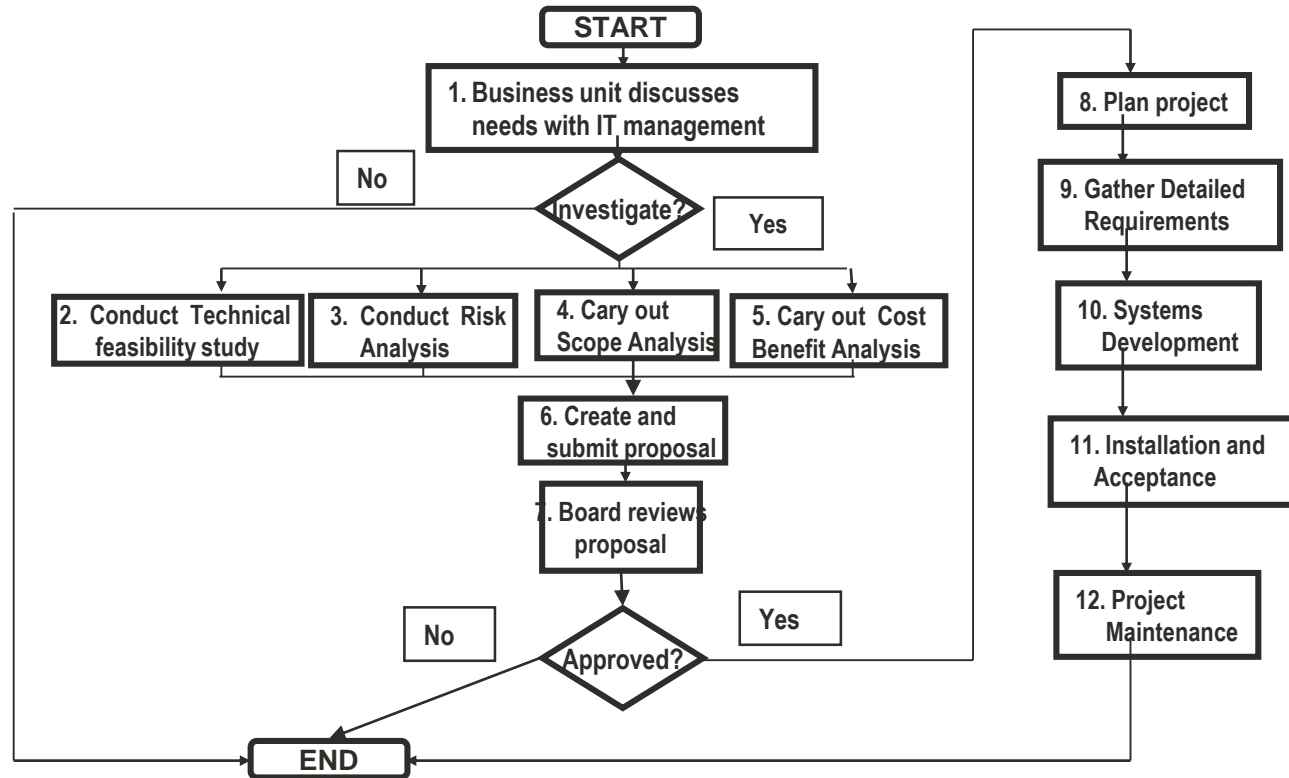
INITIATION PROCESS

Processes to define a new project or new phase of the project by obtaining authorization to start the project or phase

Purpose

- Creates a shared understanding amongst the stakeholders of the purpose and success criteria, improves deliverable acceptance, customer satisfaction and stakeholder satisfaction
- Authorises the project

INITIATION PROCESS

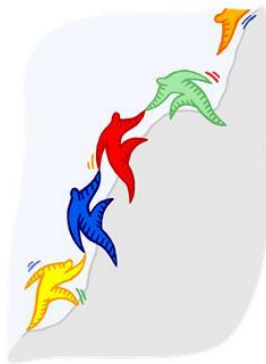


Outputs

Project Charter
Stakeholder Register

INITIATION PROCESS

A project is submitted for approval when the following have been prepared.



INITIATION PROCESS

Project Charter - A document endorsed by senior management that provides the project manager with the authority to apply organizational resources to project activities”

<input type="checkbox"/> <u>Project Charter</u>	
<input type="checkbox"/> Project Purpose	<input type="checkbox"/> Summary milestone schedule
<input type="checkbox"/> Objectives/Success criteria	<input type="checkbox"/> Summary budget
<input type="checkbox"/> High level requirements	<input type="checkbox"/> Stakeholder List
<input type="checkbox"/> High level project description and boundaries	<input type="checkbox"/> Implementation Approach
<input type="checkbox"/> High level risks	<input type="checkbox"/> Assigned PM
<input type="checkbox"/> Name and Authority authorizing the Charter	

Source: PMBOK

Project Charter

Project Title – Desktop refresh

Project Start Date : 4 Mar 2015

Project Finish Date : 4 Dec 2015

Project Manager : John Lee, jlee@ISS.com

Project Objectives : Upgrade the desktops for all employees (300) within 9 months and equip with smart devices. Includes upgrading of the wireless network to allow for 2 mobile devices per employee. Budgeted \$200k for hardware and \$100k for services

Approach

- Buy equipment from existing bulk tender contracts
- Services to be outsourced, based on 3 quotations
- Internal staff will plan and co-ordinate with the vendors and equipment suppliers

Roles and Responsibilities

Name	Role	Responsibility
Damien Soh, CEO	Sponsor	Budget and scope decisions
K.K Lai	IT Director	Provide IT resources and advisory on technical matters
J.J Lim	Project Manager	Plan and execute the project
Henry Siew	Director Corporate	Project Owner

Sign-off : (Signatures of all present)

Comments: Project must be completed by Dec 2015, due to start of business cycle in Jan 2016, *Damien Soh*

We are assuming adequate staff with the right skills are available for the project, JJ Lim
No downtime for staff while upgrades are going on, KK Lai

THE PROJECT KICK-OFF MEETING

Purpose



Marks the start of the project

Formalizes commitment of the project by all stakeholders

- Who attends?
- Key activities in a project Kick-off Meeting



DEFINING OR CONFIRMING PROJECT OBJECTIVES

The project objectives must address the following :

Problem/opportunity

Business need

Strategic fit with organizational goals

Inclusions and exclusions to scope

Products of the project

Successful completion criteria and measurements



DEFINING SCOPE

Scope Issues



How well defined is the scope?

How detailed are the requirements?

How important are the requirements?

Have the requirements been prioritized?

Can the requirements be translated to cost?

Can the requirements be implemented?

How will the requirements be implemented

“Big-bang” versus incremental

Are the requirements likely to change/grow?

Have the requirements been quantified?

SCHEDULES AND BUDGETS

Budgets and schedules may initially be “fixed”, “pre-ordained” or “non-negotiable” But these may be based on :

- Inadequate information
- Previously made promises
- Wishful thinking

As the Project Unfolds

- Detail adds costs
- Changes add costs

Initial Project Costing and Scheduling must be carefully considered

- Fixed or flexible



A Cost Benefit Analysis required to :

- Identify and quantify the benefits
- Determine if the business targets can be met

RISK

All IT projects will contain some element of risk. Project manager's duty is to minimize and manage risk

In the project initiation stage :

- Ensure possible sources of risk are not inserted in the project at this time
- Project Managers must recognize these sources of risk
- Select a project/technical strategy that will minimize risk

Stakeholder Analysis

-
- + *Organizational and management leadership*
 - + *Analysis Process*
 - + *Roles and Responsibilities*

02

STAKEHOLDER ANALYSIS - ORGANIZATIONAL MANAGEMENT AND LEADERSHIP

What is your customer enterprise personality type (Gartner):

Type A

(aggressive)

organizations are pioneers that consciously and aggressively adopt high-risk strategies to gain potentially high rewards and competitive advantage.

Type B

(mainstream)

organizations are willing to support moderate risk taking in the adoption of innovation and have the corporate skills and culture to support such initiatives.

Type C

(conservative)

organizations are cautious adopters of anything new. They are neither willing nor prepared to handle high levels of risk.

Can differ for different organizational units and corporate functions

Organization may not be in sync with market forces

Top management and staff may not always be aligned

STAKEHOLDER ANALYSIS : PROJECT STAKEHOLDERS

Who are the Project Stakeholders?

Project Stakeholders are individuals and organizations who may be actively involved in the project, and whose interests may be positively or negatively affected as a result of project execution or successful project completion.



Why do we need to consider Project Stakeholders?

Project Management is the application of knowledge, skills, tools and techniques to project activities in order to meet or exceed stakeholder needs and expectations.

Source: PMI

STAKEHOLDER ANALYSIS

To study and manage their expectations and influence (positive or negative) and their responsibilities

Carried out by a team



Assumption :
No two stakeholders
share a common
perspective of the project

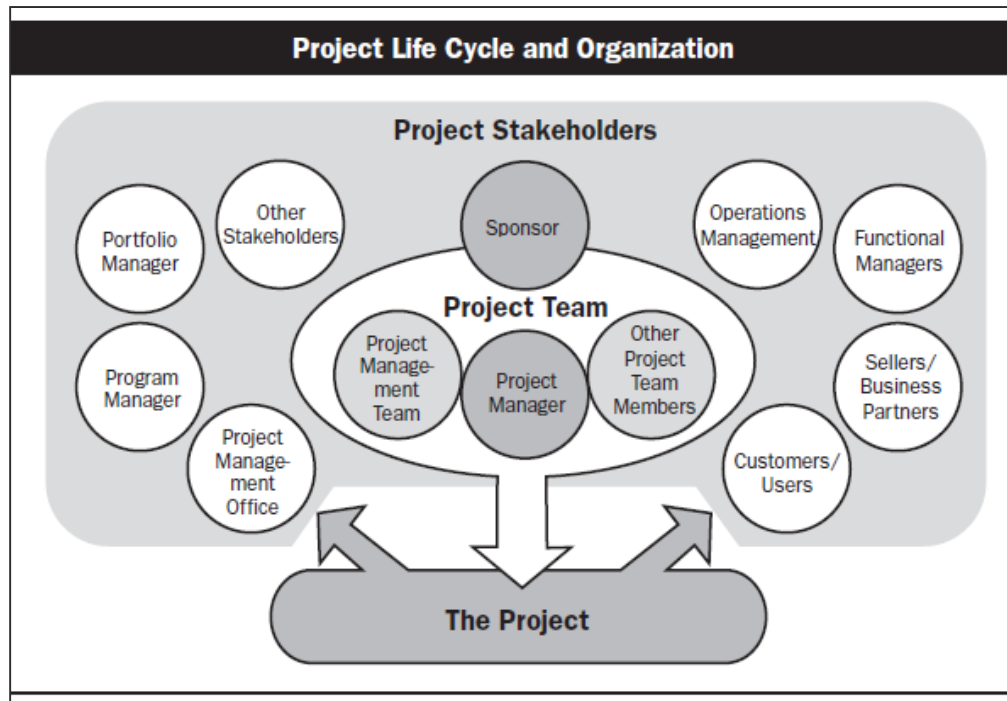
STAKEHOLDER ANALYSIS : PROCESS



STAKEHOLDER ANALYSIS : PROJECT STAKEHOLDERS

Stakeholder Identification

Who will use your system?
Who will benefit from your system?
Who will be affected by the system?
Who has an interest in your system operating?



STAKEHOLDER ANALYSIS

What information do we seek for analysis?

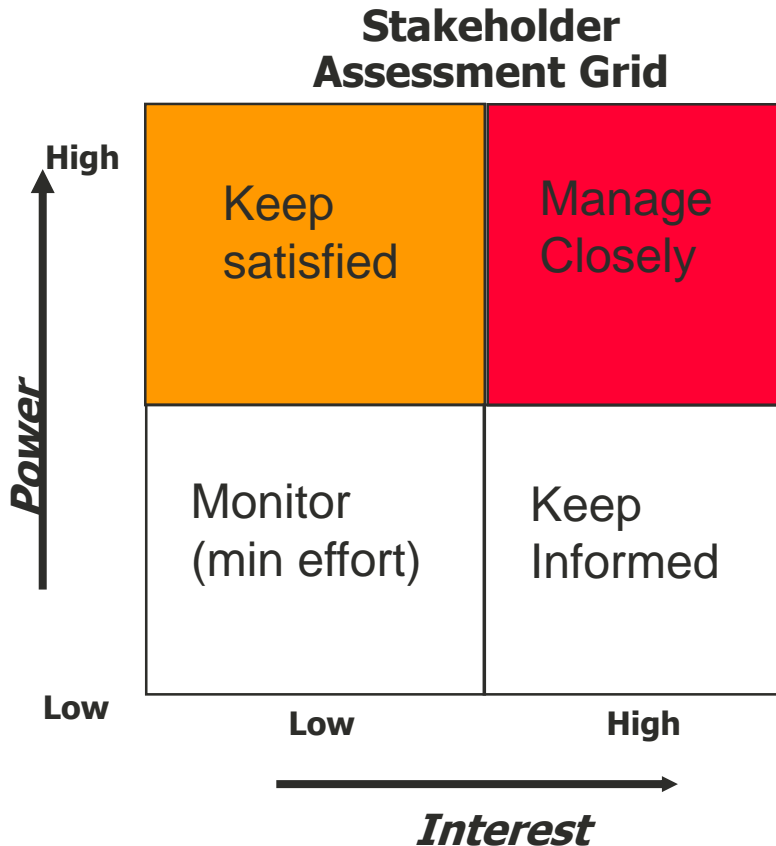
Limitations : Imperfect data means making assumptions

STAKEHOLDER ANALYSIS : CLASSIFICATION

Classification of stakeholders gives a clearer picture of who is important to the project.

Typically those who have decision making powers and those whose opinion would affect the outcomes of the project.

Stakeholders' classification may change over the life of the project



VALUE OF STAKEHOLDER ANALYSIS

In what ways does a structured analysis of the stakeholders benefit the project :

THE OUTPUTS : STAKEHOLDER REGISTER

An example

The register consolidates information about stakeholders from various sources and perspectives.

It's used for the analysis and subsequent formulation of engagement activities to address the needs of the stakeholder.

This document is highly sensitive and must be treated in-confidence

	Director IT
Roles & Responsibilities in organization and project	policy maker on IT matters in company Steering committee member for IT
Key Concerns	Desires a smooth replacement of the equipment and upskilling of the IT staff in supporting mobile devices. Must keep to the budget
Impact to project	High
Project Priority	High
Response Action to Stakeholder's Key Concerns	Regular updates on the budget utilization. Survey and feedback from the company on how on the process and improvements Training plan review from the IT staff

STAKEHOLDERS ROLES

The Project Manager

Project Steering Committee

The Project Sponsor

The Project Owner

The user manager

The vendor project manager

The SCRUM Master

The Product Owner

STAKEHOLDERS ANALYSIS : ROLES AND RESPONSIBILITIES



The Project Manager

- Direct and co-ordinate
- Monitors progress
- Determines a project management approach, including a project strategy and an appropriate lifecycle and Methodology
- Prepares a Project Plan
- Regular reporting to committee
- Keeps good records of project meetings
- Manage Vendor (if appropriate)



Project Steering Committee

- Oversee overall implementation of project
- Report to top management on progress of project
- Provide overall policy direction and endorses priorities
- Approves major changes to requirements
- Approve variations to project objectives, scope, approach and schedule
- Ultimate reviewer and approver of all deliverables of the Working Committee
- Endorses System Acceptance



The Project Sponsor

- Provides the justification and executive support for the project
- Usually involved at the inception of the project
- Will provide high-level (but not necessarily consistent) support

STAKEHOLDER ANALYSIS : ROLES AND RESPONSIBILITIES



Project Owner

- Owner of the system
- May be (but also may not be) a User manager
- May be non –technical
- Involved in operation/ maintenance of system
- Will not always appreciate this role!



User Manager

- Provide domain knowledge and user requirements
- Work within the User organization to gain the necessary support and acceptance of the project
- Ensure the requirements defined meet business needs
- Validate the requirements have been met by
- *Conducting Reviews*
- *Performing User Acceptance Tests*



Vendor Project Manager

- Gather user requirements
- Design the technical architecture for their provided components
- Develop the system according to the requirements specified
- Perform Unit and System Integration Tests
- Integrate components into overall system

Implementation Strategy

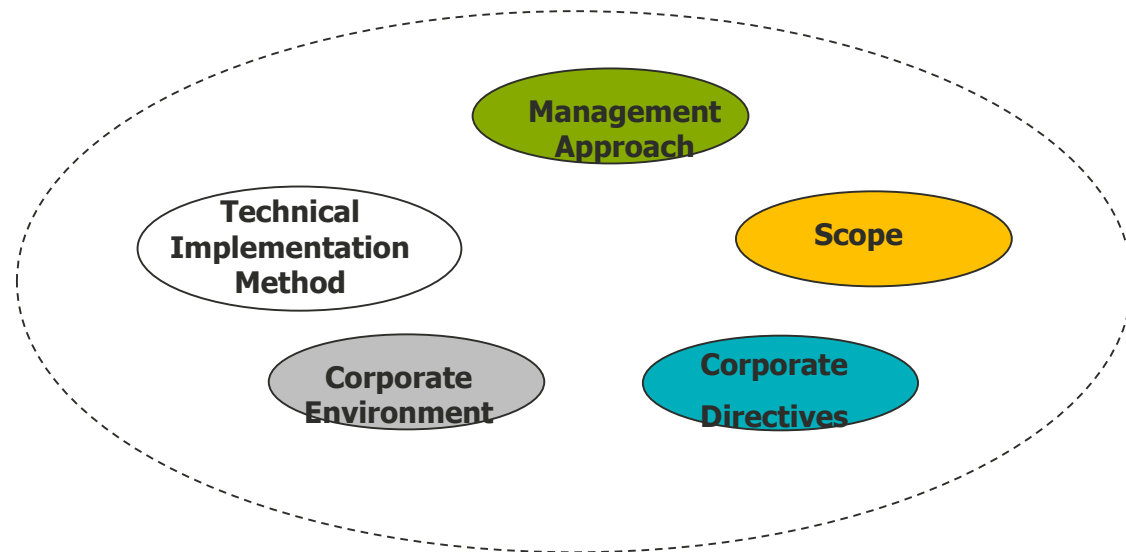
-
- + *Organizational and Management Leadership*
 - + *Implementation Strategy*
 - + *Technical Strategy*

03

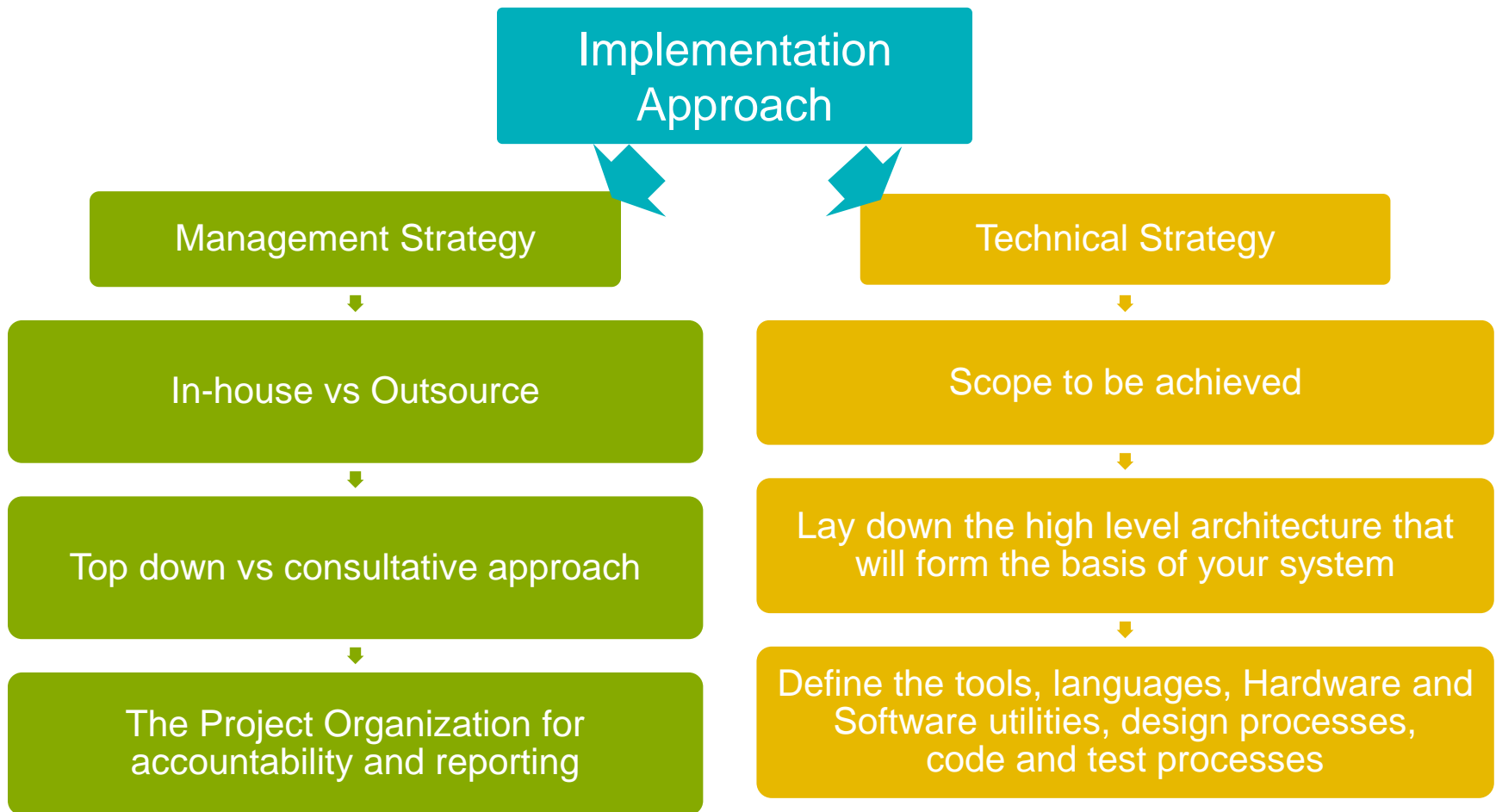
WHAT IS AN IMPLEMENTATION APPROACH?

It describes :

- How you are going to organize and manage the project
- What the project products are
- How you are going to develop the project products
- Who will perform the project



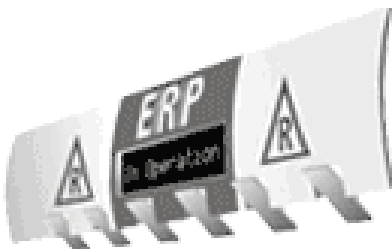
WHAT DOES AN IMPLEMENTATION APPROACH COMPRISE OF?



IMPLEMENTATION APPROACHES CAN BE VERY DIFFERENT

An ERP Implementation

- Large formal user involvement at several levels
- Formal weekly (monthly) progress meetings and reports
- Rolling delivery of several different functions
- Formal requirement specification and review and agreement
- Fixed technology approach



An AGILE project

- Frequent (daily) progress meetings, but no formal reporting
- Requirements are flexible
- Delegation of technical decision making to individual members
- Adoption of “sprints “ to create iterations of work products
- Users involved in development, decision-making and compromising

POSSIBLE IMPLEMENTATION APPROACHES

(NOT Exhaustive)

COTS versus Bespoke

System based versus SaaS

In-house versus Outsource

Incremental versus Big Bang

Evolutionary versus Standard

Open Source versus Proprietary

Agile versus Conventional Methodology

Standards Based versus One-of-a-Kind

Generic versus Proprietary e.g. Rational Unified Processes...

CRITERIA FOR SELECTING AN IMPLEMENTATION APPROACH

Guiding questions

- How big is the project?
- How is the customer working with you
- How experienced, capable and flexible are your staff?
- What are the maturities of the technologies you are using?
- What does your senior management expect of you?
- How much are You willing to delegate?
- How many organizations are collaborating?
- Are you outsourcing? Local or overseas?



IN-SOURCING/OUTSOURCING/BEST-SOURCING

S	W
O	T

When do you consider Outsourcing?

Corporate Policy
Lack of staff resources
Focus on core competencies

Cost savings
Vendors have demonstrated their capabilities
Risk
No strategic benefits

If the application gives strategic advantage, and

It is important to control and develop this application,
You believe that your organization needs to develop and exploit this technology

→ *In-source*

If the application gives strategic advantage, but

You may or may not have the capability to control and develop this application,
But you believe that your organization needs to exploit this technology

→ *Best-source*

If the application gives immediate, but

non-critical support and
You do not have the capability or desire to control and develop this application,
Your organization regards this as a “utility”

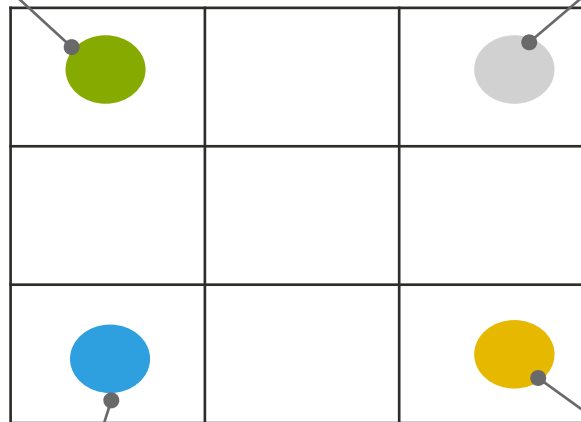
→ *Out-source*

CHOOSING THE RIGHT OUT-SOURCING PARTNER

In the following circumstances :

Contract out the required services to the most cost effective vendor, or buy an existing product

Technology Maturity



Create your own integration team, but contract a development by the most highly qualified Vendor who is also willing to form a long-term partnership with you

Need for integration with existing infrastructure

Contract a development by the most highly qualified Vendor

IMPACT OF IMPLEMENTATION APPROACH

The consequence of the selected implementation approach would affect :

Size and Composition of Project Team

Feasible Solution Space and Tradeoffs between:

- Requirements and Specifications
- Project Schedule
- Cost and Resource Requirements
- Quality Achievable

THE BOTTOM LINE

The selected implementation approach :

- Should match the Characteristics of the Organization and the Problem Domain
- Major Tradeoff between the Project Team's Comfort Zone and Appropriateness for the Project
- How do you Bridge the Gap?

AN EXAMPLE: BRING IN A CONSULTANT

PWC (PriceWaterhouseCoopers) provides four fundamental reasons why organizations should bring in outside experts:

People - Access to specialist skills or additional labor

Process - To use a tried-and-tested methodology

Perspective - To get independent or innovative input

Politics - To validate a decision or push through an unpopular change

Expert advice and help you receive in these situations can save you time, dollars, effectiveness, momentum, and peace-of-mind (for the bosses)

But Not Always!

ANOTHER EXAMPLE: MAKE USE OF BEST PRACTICE CHECKLISTS

Assessment Checklist

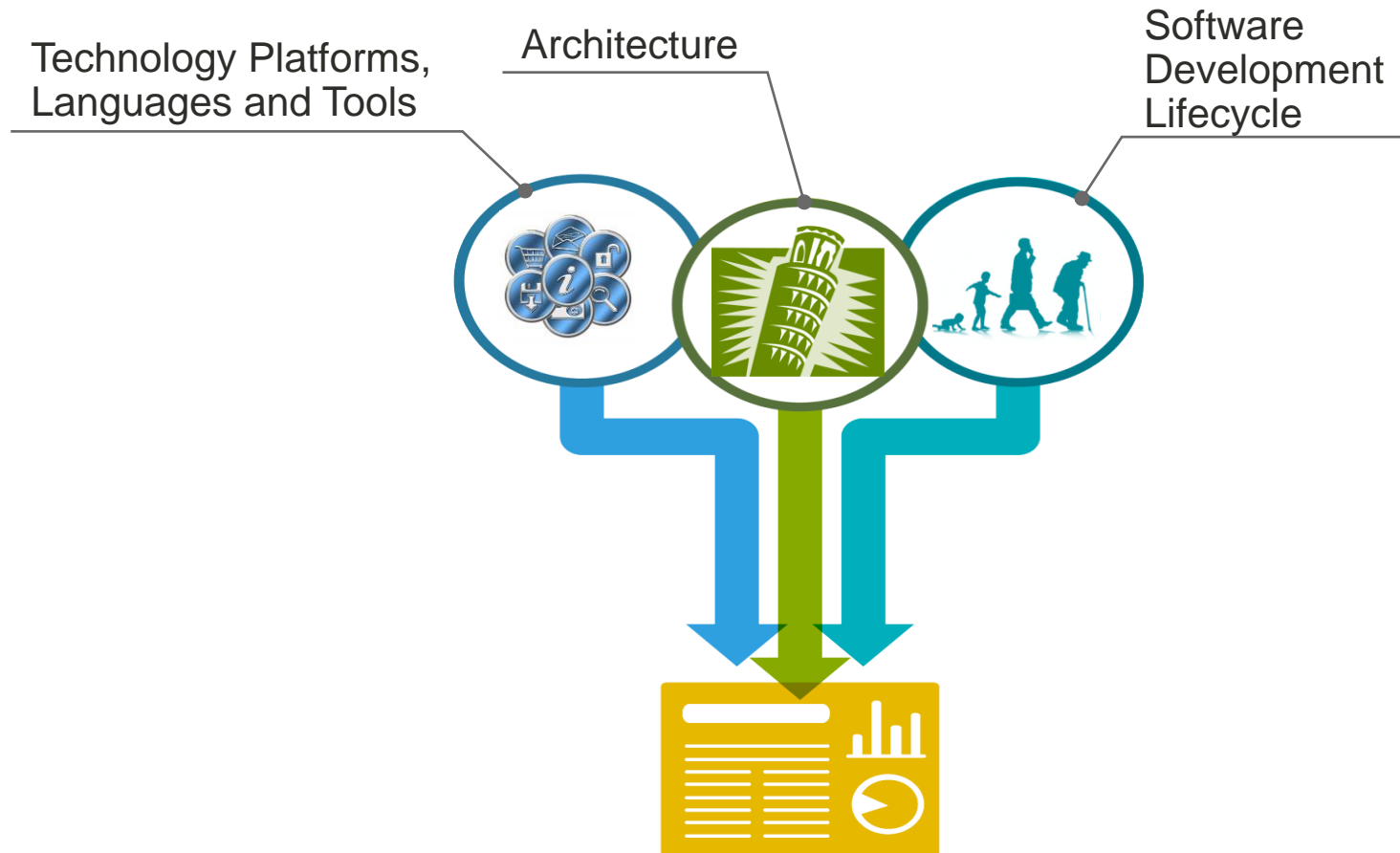
See Vendor Selection: A Fourteen-Point Guide adapted from Datamation
@<http://itmanagement.earthweb.com/netsys/article.php/3647621/Vendor-Selection-A-Fourteen-Point-Guide.htm>

Technology Strategy

-
- + *Platform*
 - + *Architecture*
 - + *Development Lifecycle*

02

WHAT IS A TECHNOLOGY STRATEGY?



CRITERIA IN SELECTING A TECHNOLOGY STRATEGY

Technical solution may already be pre- determined

Enhancement/Upgrade of
existing system

Corporate Policy

Corporate Initiative

Results of Prototyping

Interfacing/Interacting
with other systems

Required functionality

Novelty of functionality
and ease of
implementation

Non-functional requirements

Performance

Reliability and availability

Security

Scalability

Usability



TECHNOLOGY PLATFORMS, LANGUAGES AND TOOLS (NOT EXHAUSTIVE)

Platforms

PC, Mobile, Mainframe,
Mini, Parallel, Fault
Tolerant, Smart Devices
J2EE, .NET, Client
Server, LAMP Stack,
CICS Transaction Server

Languages

Java, C#, Objective-C,
Python, Ruby, PB,
COBOL, ADA, DSLs
(Domain Specific
Languages)

Tools

Software Development
Environments,
Configuration
Management,
Code Generators
Test Generators
CASE Tools

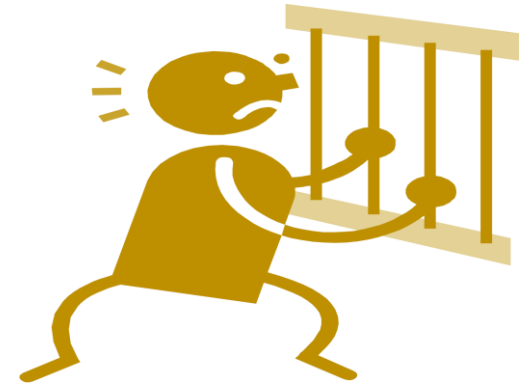
CONSTRAINTS IN SELECTING A TECHNOLOGY PLATFORM

Cost and schedules

- Costs of adopting technologies
- Cost of maintenance and support

Corporate and individual knowledge and experience

Risks created by adopting certain technologies or implementation paths



IMPACT OF TECHNOLOGY PLATFORM

Required Skill Sets and Staff Recruitment

In-House Staff Training and Competency Development

Long Term Support, Maintenance and Sustainability Issues

Need for (even) more Consultants and Experts

THE BOTTOM LINE

When choosing a Technology Platform remember

There are no Silver Bullets

D-I-Y is over-rated

There is Safety in Mainstream

DEFINITION: ARCHITECTURE

The architecture of an IT system is the highest level description of its components, interrelations, and functions



- Major hardware and software components
- Physical and logical linkages between components
- Distribution of functions across components

Selecting and specifying an appropriate architecture is the first stage of design

ARCHITECTURE

(NOT Exhaustive)

Object Oriented Three Tier Layered Internet Architecture

Legacy database centric batch processing

KE type systems (Intelligent agents, rule-based, fuzzy logic, genetic algorithm, cased based reasoning, neural networks)

Real time event action systems e.g. embedded process control

Wireless mobile

Client side mash-ups

Massively parallel processing

Dataflow Pipeline (pipes and filters)

Distributed communicating processes

Blackboard / content addressable memory repositories

MAJOR CONSIDERATIONS IN SELECTING AN ARCHITECTURE

Guiding Questions

- Will the system be *distributed*?
- Do different components perform substantially different functions (heterogeneous system)?
- Are *the right functions assigned to the right components*
 - Self-contained functions (execute entirely on one component)
 - Distributed functions
 - Response-time critical functions
 - Background functions
- How are components linked together?
- What must be communicated between components, and by what means?
- What external systems must interoperate with our system?

MAJOR CONSIDERATIONS IN SELECTING AN ARCHITECTURE (CONT.)

Private and Public Interfaces

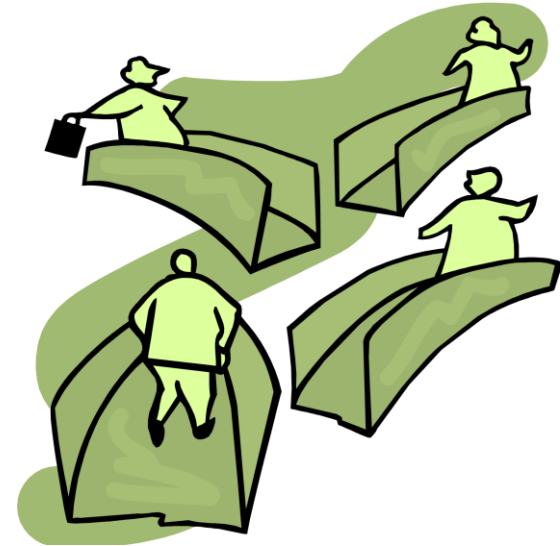
Interfaces to third parties are typically public

A fixed standard to which we must conform

Interfaces within our system are private

We can design them as we please

However, it is increasingly the state of the practice to use standard interfaces whenever possible, based on technologies such as XML.



MAJOR CONSIDERATIONS IN SELECTING AN ARCHITECTURE (CONT.)

Non functional requirements

Performance

Required speed of transactions

Data storage requirements

Loadings

Reliability and availability

Accuracy

Levels of Service

Security

Scalability

Interoperability

Reusability

Maintainability

Portability

Usability

All of these should be specified in the system requirements specification or the Service Level Agreement

AN EXAMPLE

Prototype

Prototyping is a learning experience. Its value lies not in the code you produce, but in the lessons you learn.

Things to prototype

Architecture

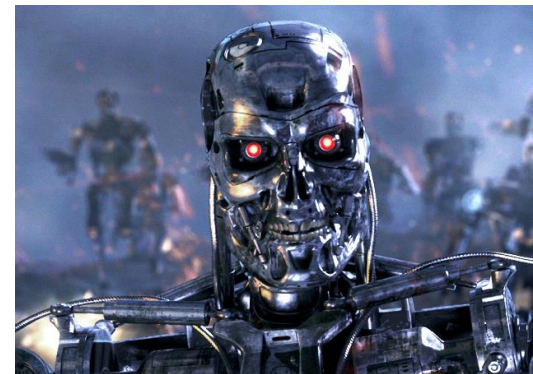
New functionality in an existing system

Structure or contents of external data

Third-party tools or components

Performance issues

User interface design



THE ROLE OF PROTOTYPING

The initiation or feasibility analysis stage is often the place where we identify a need to prototype

Novel functions require a proof of concept

Unfamiliar interfaces require a proof of interoperability

Strict quality requirements require a proof of performance

In general, a prototype answers a key question about our proposed technology platform and architecture



WHAT IS A SOFTWARE DEVELOPMENT LIFE CYCLE?

What does a Software Development Life Cycle (SDLC) do?

It defines a specific approach to producing software

Why do we need an SDLC?

Complex processes are needed to develop complex products

The nature and composition of software products is not generally known at the start of the project

It only unfolds during the project progress

Components of an SDLC

A set of descriptions of each software development stage

A definite order in which these stages are executed

Transition criteria from one stage to another

Defines the deliverables to be produced during a project

Reference the standards required by a project

WHAT IS A SOFTWARE DEVELOPMENT METHODOLOGY?

A Software Development Methodology is

A set of techniques and tools used for carrying out the tasks or phases in the SDLC

The toolset within each methodology have associated

- Notations
- Diagrams
- Terminology
- Rules

Can usually be used in different lifecycle models

The overall goal of a software methodology is to produce systems that are

Of high quality i.e. meet users requirements

Easy to understand

Reliable

Flexible

Easily maintainable

Progressively developed

Minimize lifecycle costs

EXAMPLES OF SDLC'S

Generic SDLC Models

- Waterfall Models
- Incremental Models
- Evolutionary Models
- Adaptive Models

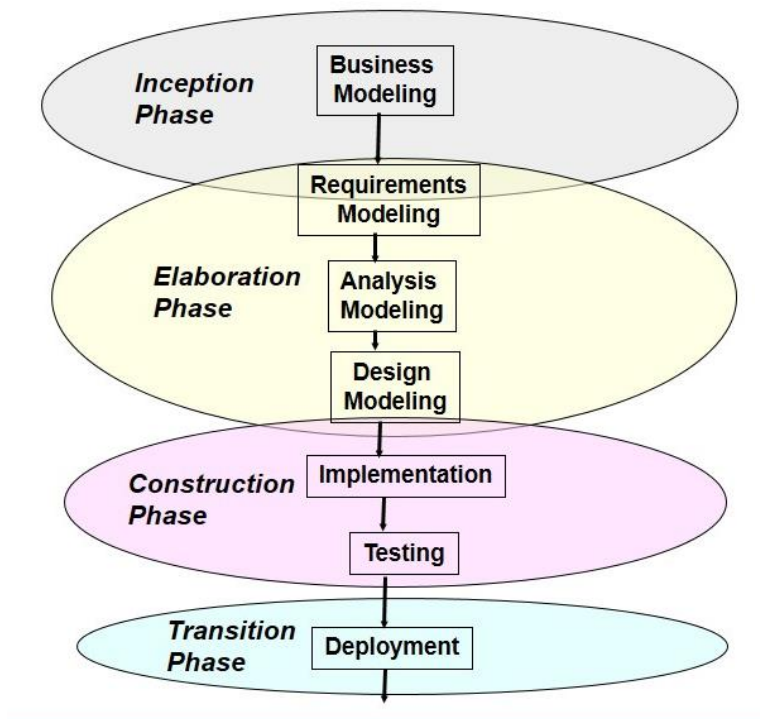
Examples of Technological/ Application specific life cycles

- ERP
- RUP
- Workflow application development



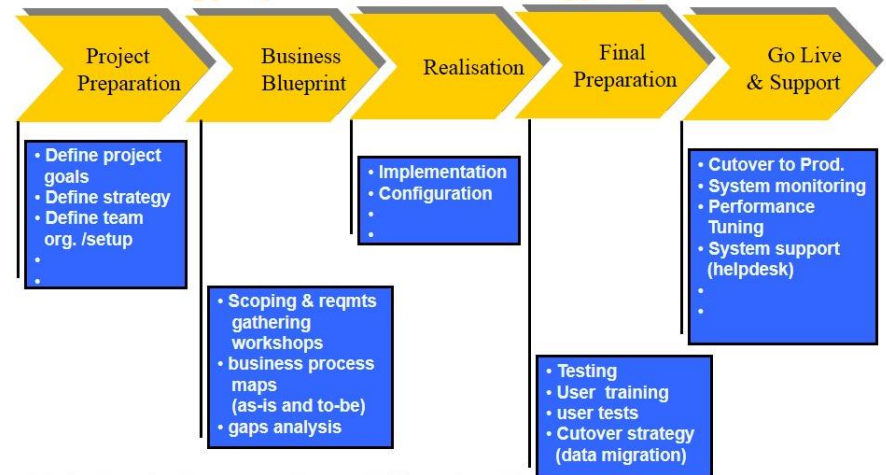
EXAMPLES OF TECHNOLOGICAL/APPLICATION SPECIFIC LIFE CYCLE

RUP *Examples of Technological/Application specific*



ERP

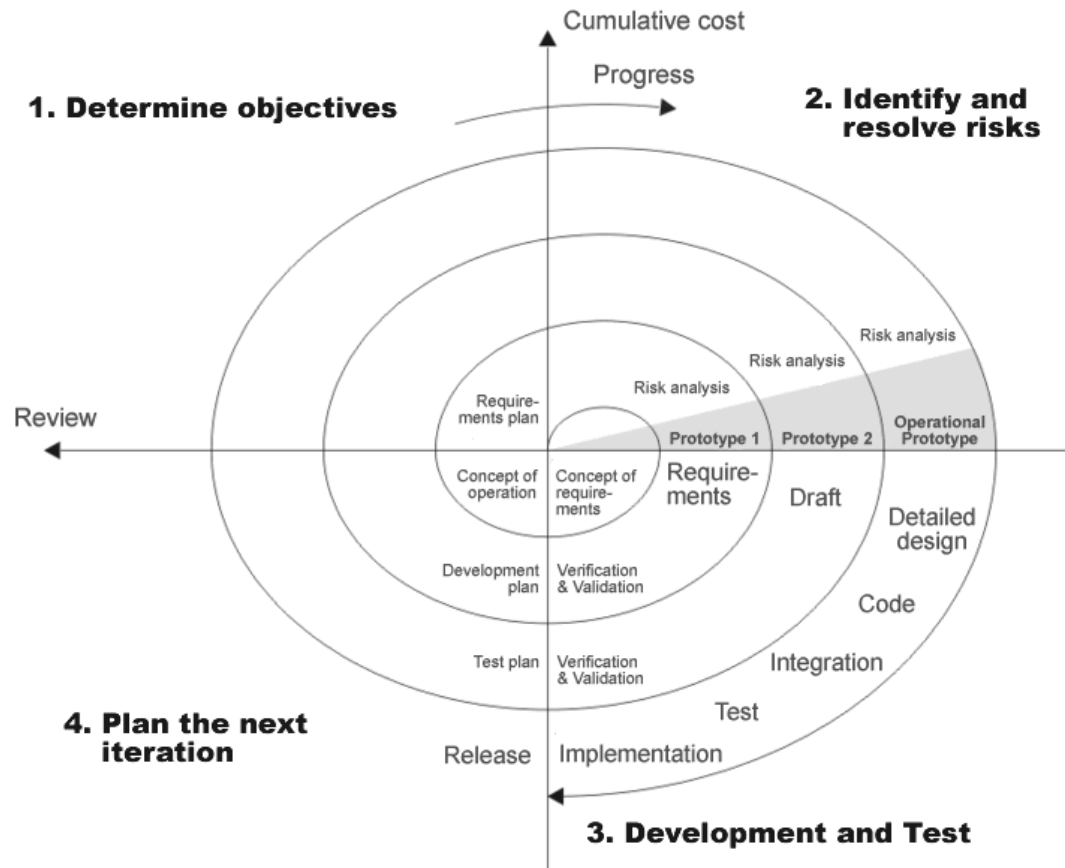
Technology Specific Methodology eg. ASAP



*End of each phase must have a “client signoff”

BOEHM'S SPIRAL MODEL - EXAMPLE OF RISK LOCALISATION

Special case of evolutionary software development, produce software in series of prototypes



BOEHM'S SPIRAL MODEL - EXAMPLE OF RISK LOCALISATION

Special case of evolutionary software development, produce software in series of prototypes

Lifecycle:

- Determine objectives, alternatives, constraints
- Evaluate alternatives, identify and combat Risks
- Develop prototype
- Plan next phase

Advantages of incremental development

- Risk driven not life-cycle driven
- Maximizes re-use
- Rapid development of prototypes provides increasing functionality

Disadvantages of Spiral Model

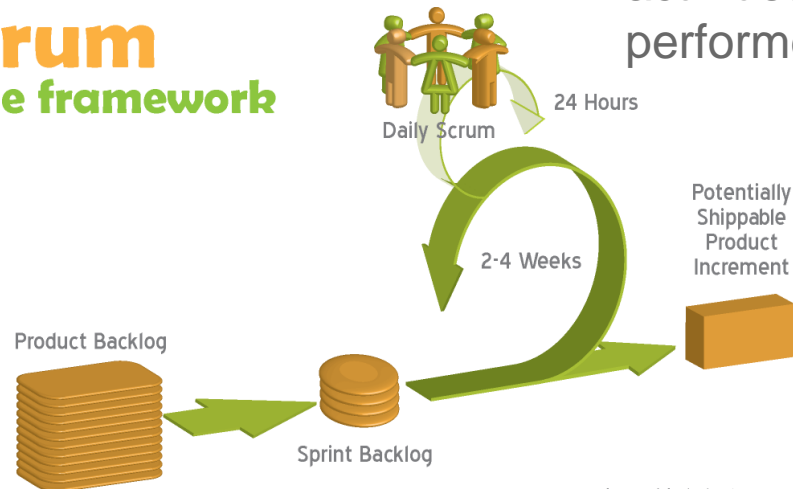
- Emphasis on prototypes not functional releases
- Long development schedule

ADAPTIVE LIFE CYCLES

Also known as change driven or agile methods. E.g. SCRUM

- Each iteration is short (usually 2 to 4 weeks) with fixed time and cost
- Respond well to high levels of change and ongoing stakeholder involvement
- Overall scope decomposed into prioritized requirements (or features) in the Product Backlog

Scrum the framework



- In each iteration – all PM process group activities will be performed

- At start of each iteration (or Sprint), determine how many of highest priority items on the product backlog can be delivered (Sprint Backlog). Delivered it. Customer reviewed it. Repeat this step

DECISIONS TO BE MADE IN ADOPTING YOUR IMPLEMENTATION APPROACH AND TECHNOLOGY STRATEGY

You must decide how you wish to build the system

In multiple releases

In one big release

What form will your system implementation take?

What SDLC will be used

What (if any) methodology will be used?

Who will do the implementation?

In-house, Outsource, Best-source

How will the implementation be performed ?

Bespoke development,

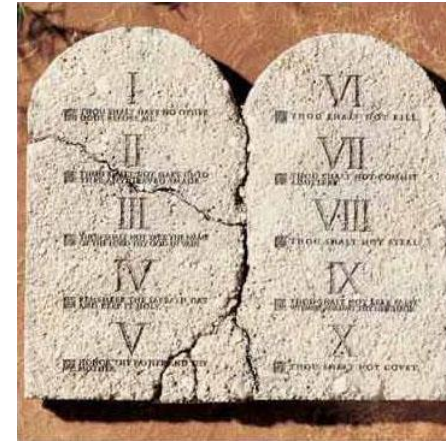
Customisation of an existing application,

Utilising existing components to provide major functionality from in-house sources ,Commercial packages, Open Source Software, new functionality from scratch.

SUMMARY: THE IMPLEMENTATION APPROACH AND TECHNOLOGY STRATEGY

Describes

- What you are going to build
- How you are going to build it
- And who will build it
- It will describe your management approach
- It will define the major phases and the lifecycle to be used in the development of your system
- It will form the basis of the high level architecture and design of your system
- It will identify the tools and processes that will be used
- It will identify how the system will be built



blog.bincsearch.com

HOW DO YOU FORMALIZE OR DOCUMENT YOUR IMPLEMENTATION APPROACH AND TECHNOLOGY STRATEGY?

Project plan

- Organizations, Schedules, and lifecycles

Quality Plan

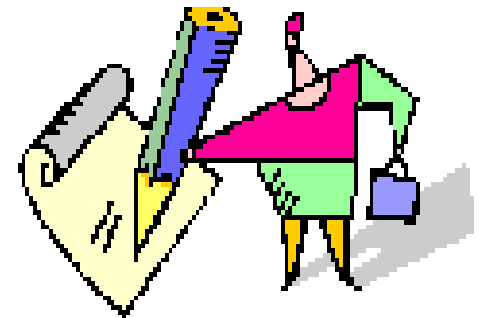
- Roles and Responsibilities, lifecycles, methods and processes

Other Quality documents, e.g.

- Coding Standards
- Design methodology descriptions

Technical Strategy Description

High level Architecture



AT THE END OF THE DAY

IMPLEMENTATION APPROACHES

Don't Be a Slave to Formal Methods

Don't blindly adopt any technique without putting it into the context of your development practices and capabilities.

Critically Analyze What You Read and Hear

Don't be swayed by vendors, media hype, or dogma. Analyze information in terms of you and your project.

Invest Regularly in Your Knowledge Portfolio

Make learning a habit.

TECHNOLOGY STRATEGY

How to choose a Technology Strategy

Deciding on any software is still an art

We have to spend time to gather information & understand the software

- Heuristic
- Strategy versus execution
- Agility
- Support gives peace of mind (like insurance) and throat to choke (even if bug fix might not be immediate)

Being a good (and more importantly experienced) Project Manager helps