BITS Pilani

Pilani | Dubai | Goa | Hyderabad

# Software Project Management

## SSZG622

Prof. Narasimha Bolloju, Ph.D.
Department of Computer Science and Information Systems
Birla Institute of Technology and Science, Pilani

# About the SPM course

- Flipped mode with recorded lecture content on Taxila/elearn platform
- Small group work (on MS-Teams)
- ACL case study used in all contact sessions

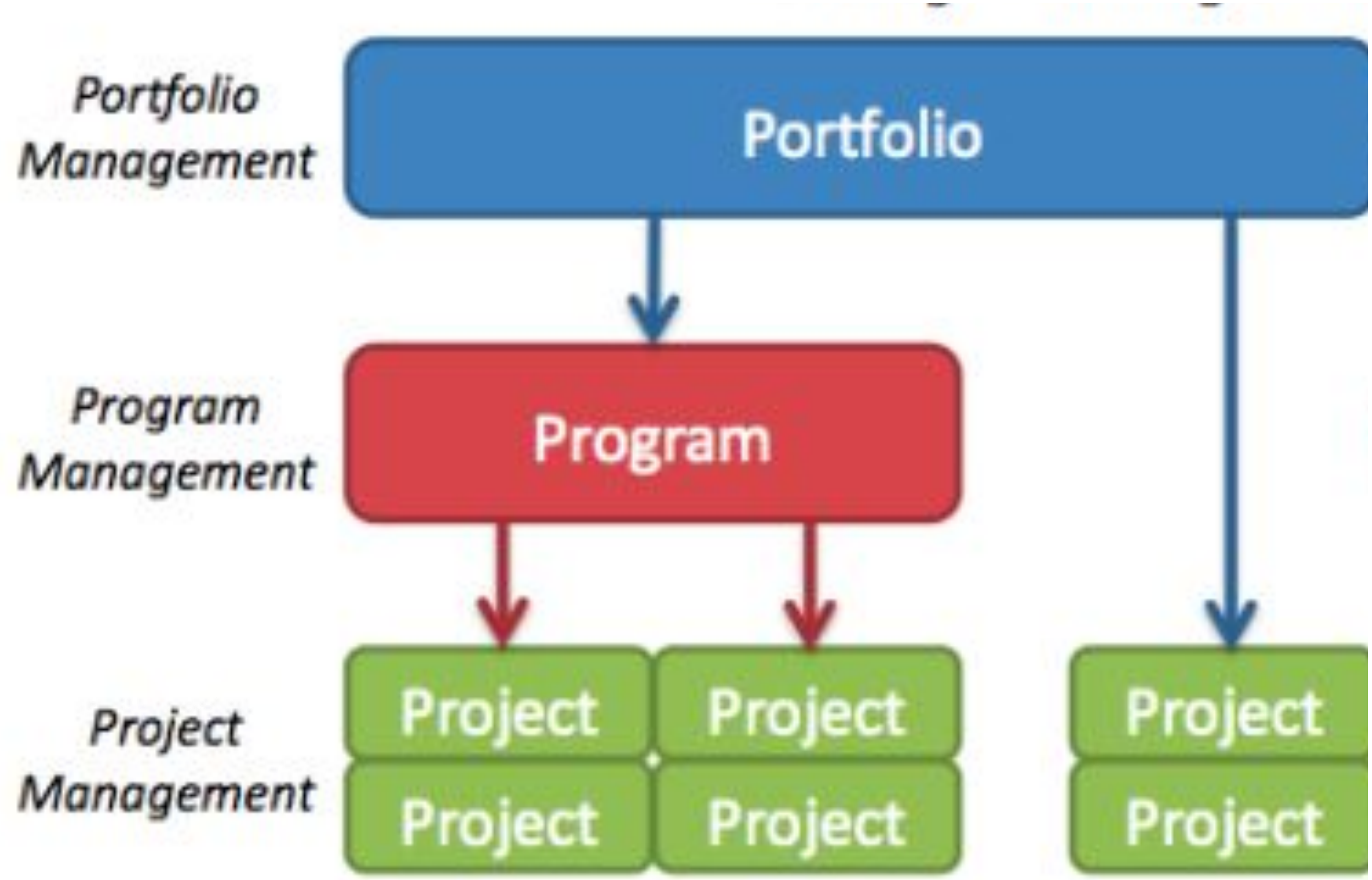
# Activity - CS01#1

a)What do software project managers do or expected to do?

b)What are the typical challenges associated with managing software development projects?

# Topics

- Project vs. program vs. portfolio
- SDLC vs. project life cycle
- Why SPM is challenging?
- Evaluation scheme and assignment details
- Objectives for information system and project management

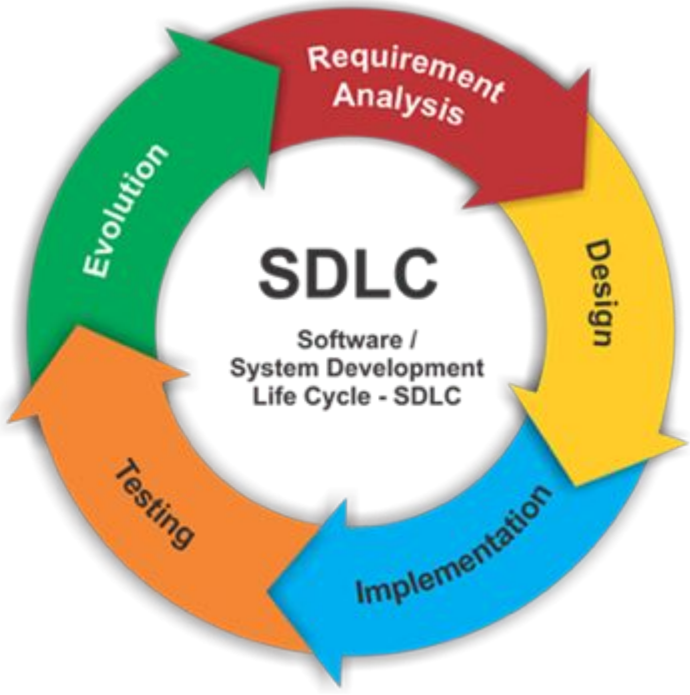


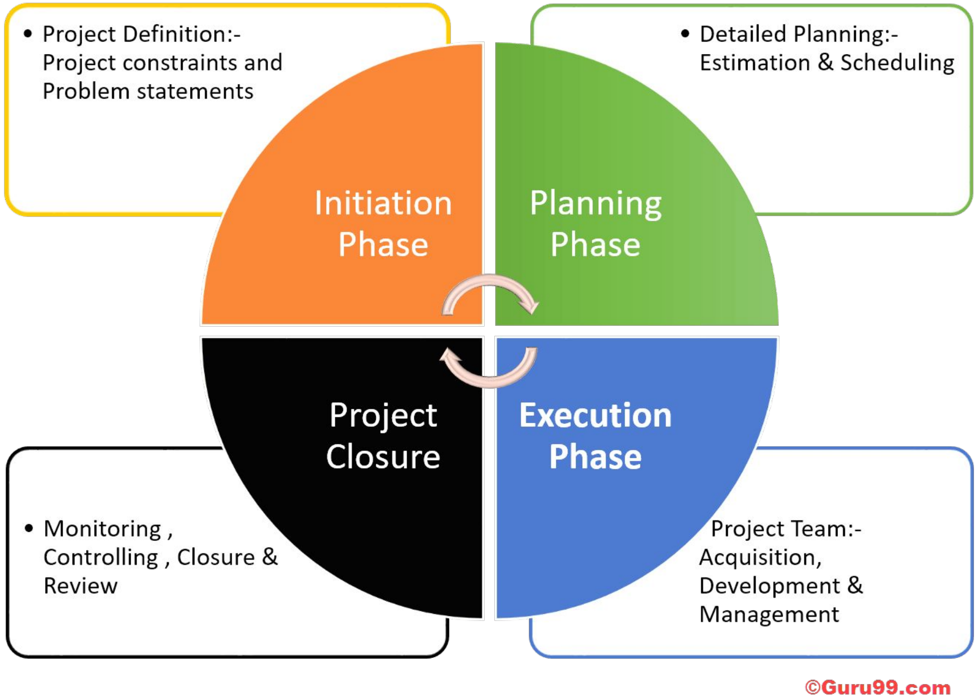
https://kevinberardinelli.com/2011/03/07/project-management-body-of-knowledge-pmbok-notes-introduction-chapter-1/

# Activity - CS01#2

Why A2Z Cinemas Ltd (ACL) wanted to start Mux Core Project (especially when they have other systems providing the required functionality)?

# SDLC vs project life cycle

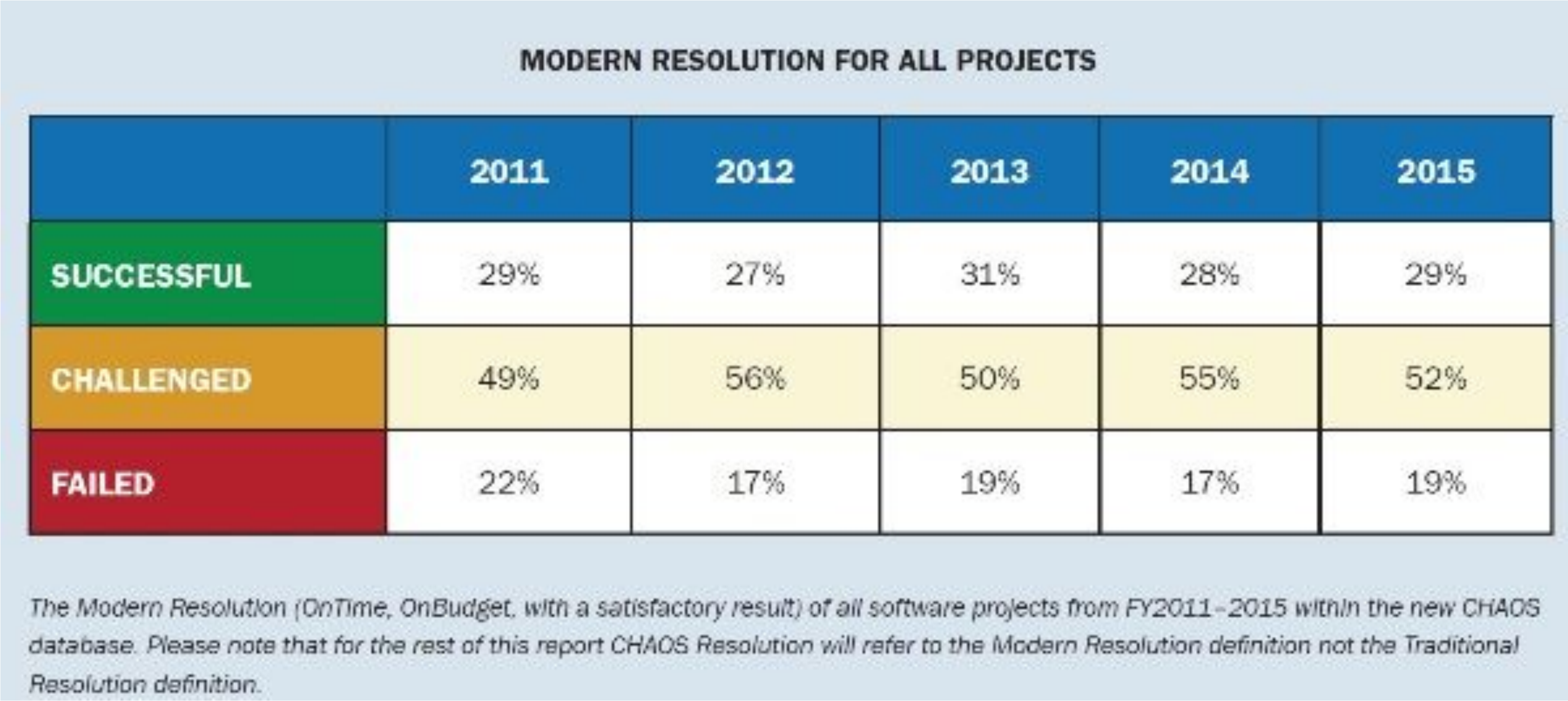




- Project Definition:- Project constraints and Problem statements
- Detailed Planning:- Estimation & Scheduling
- Monitoring , Controlling , Closure & Review

Project Team:- Acquisition, Development & Management

# Software development projects – Success vs failure

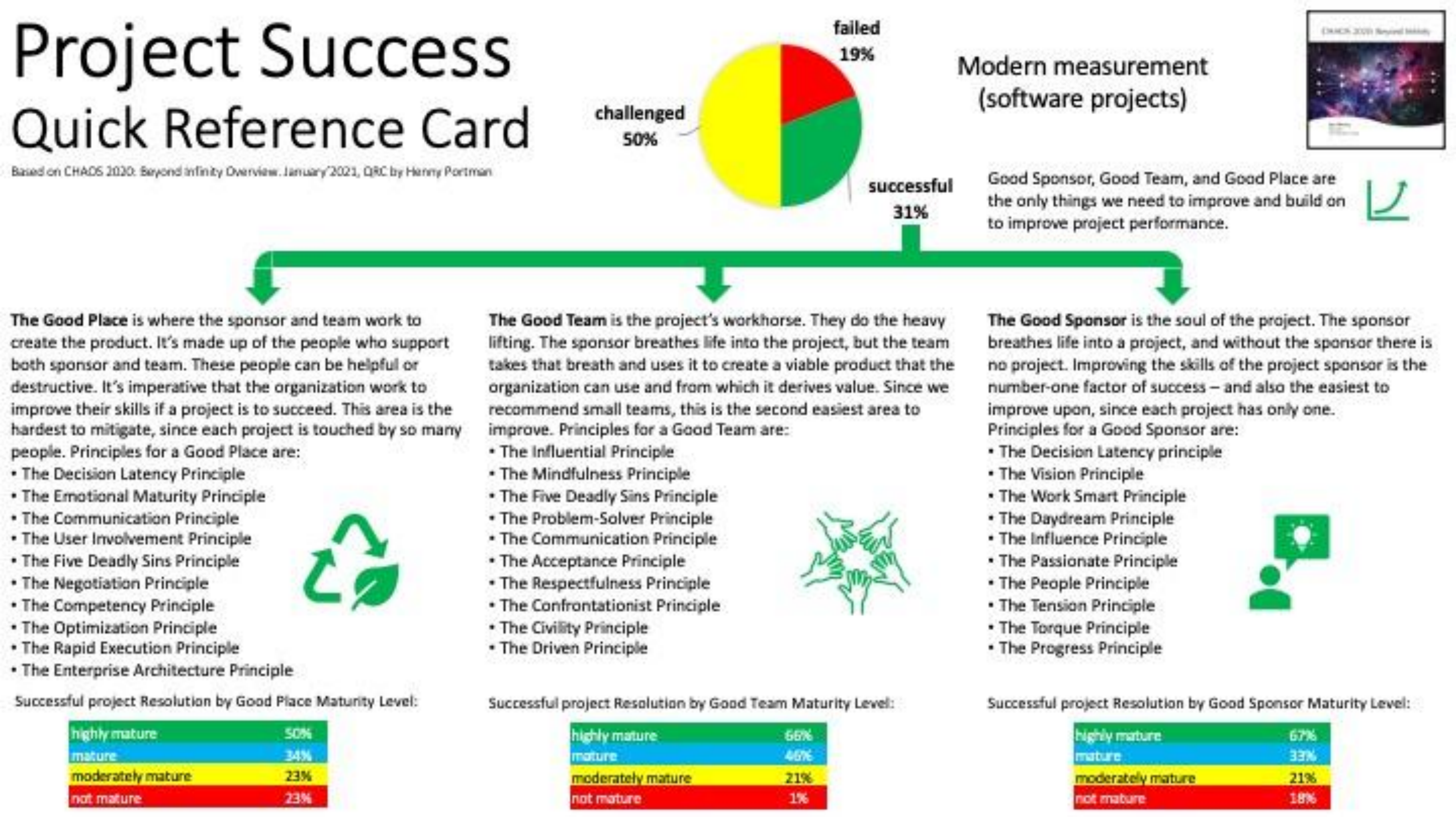


MODERN RESOLUTION FOR ALL PROJECTS

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

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

https://www.infoq.com/articles/standish-chaos-2015



# Project Success
## Quick Reference Card

Based on CHAOS 2020: Beyond Infinity Overview. January'2021, QRC by Henny Portman

failed
19%
challenged
50%
successful
31%

Modern measurement (software projects)

Good Sponsor, Good Team, and Good Place are the only things we need to improve and build on to improve project performance.

**The Good Place** is where the sponsor and team work to create the product. It's made up of the people who support both sponsor and team. These people can be helpful or destructive. It's imperative that the organization work to improve their skills if a project is to succeed. This area is the hardest to mitigate, since each project is touched by so many people. Principles for a Good Place are:

- The Decision Latency Principle
- The Emotional Maturity Principle
- The Communication Principle
- The User Involvement Principle
- The Five Deadly Sins Principle
- The Negotiation Principle
- The Competency Principle
- The Optimization Principle
- The Rapid Execution Principle
- The Enterprise Architecture Principle

Successful project Resolution by Good Place Maturity Level:

| | |
|---|---|
| highly mature | 50% |
| mature | 34% |
| moderately mature | 23% |
| not mature | 23% |

**The Good Team** is the project's workhorse. They do the heavy lifting. The sponsor breathes life into the project, but the team takes that breath and uses it to create a viable product that the organization can use and from which it derives value. Since we recommend small teams, this is the second easiest area to improve. Principles for a Good Team are:

- The Influential Principle
- The Mindfulness Principle
- The Five Deadly Sins Principle
- The Problem-Solver Principle
- The Communication Principle
- The Acceptance Principle
- The Respectfulness Principle
- The Confrontationist Principle
- The Civility Principle
- The Driven Principle

Successful project Resolution by Good Team Maturity Level:

| | |
|---|---|
| highly mature | 66% |
| mature | 46% |
| moderately mature | 21% |
| not mature | 1% |

**The Good Sponsor** is the soul of the project. The sponsor breathes life into a project, and without the sponsor there is no project. Improving the skills of the project sponsor is the number-one factor of success – and also the easiest to improve upon, since each project has only one. Principles for a Good Sponsor are:

- The Decision Latency principle
- The Vision Principle
- The Work Smart Principle
- The Daydream Principle
- The Influence Principle
- The Passionate Principle
- The People Principle
- The Tension Principle
- The Torque Principle
- The Progress Principle

Successful project Resolution by Good Sponsor Maturity Level:

| | |
|---|---|
| highly mature | 67% |
| mature | 33% |
| moderately mature | 21% |
| not mature | 18% |

https://hennyportman.wordpress.com/2021/01/06/review-standish-group-chaos-2020-beyond-infinity/

# Activity - CS01#3

What are the possible reasons for unsuccessful projects (i.e., failed or challenged projects)?

# Evaluation scheme

| No | Name | Type | Duration | Weight | Day, Date, Session, Time |
|---|---|---|---|---|---|
| EC-1 | **Quiz 1** | Online | 1 week | 5% | Before mid-sem test |
| | **Quiz 2** | Online | 1 week | 5% | After mid-sem test |
| | **Assignment 1 and Assignment 2** | Report | ~ 2 weeks each | 10% + 10% | Before the mid-sem test and Before the comprehensive exam |
| EC-2 | **Mid-Semester Test** | Closed Book | 2 hours | 30% | Refer to the semester calendar |
| EC-3 | **Comprehensive Exam** | Open Book | 2 hours | 40% | Refer to the semester calendar |

# Individual assignments

- You will prepare and submit a report on a topic related to software project management in two parts
- You are expected to put in 10 to 12 hours of total effort in study, collecting relevant information, and preparation of the report
- Submissions will be checked for plagiarism using Turnitin tool (no marks will be awarded for reports with similarity score above 25%)
- Further details to be provided after topic assignment in next week

# Sample topics for the assignment

| |
|---|
| Software effort estimation |
| Project initiation or planning |
| Project management life cycle |
| Project risk management |
| Techniques for quality improvement |
| Project scheduling |
| Build or buy decision making |
| Project evaluation |

# Activity - CS01#4

For the Mux-Core project of ACL, identify objectives for

a) Mux-Core system and

b) Mux-Core project management

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

# SS ZG622:

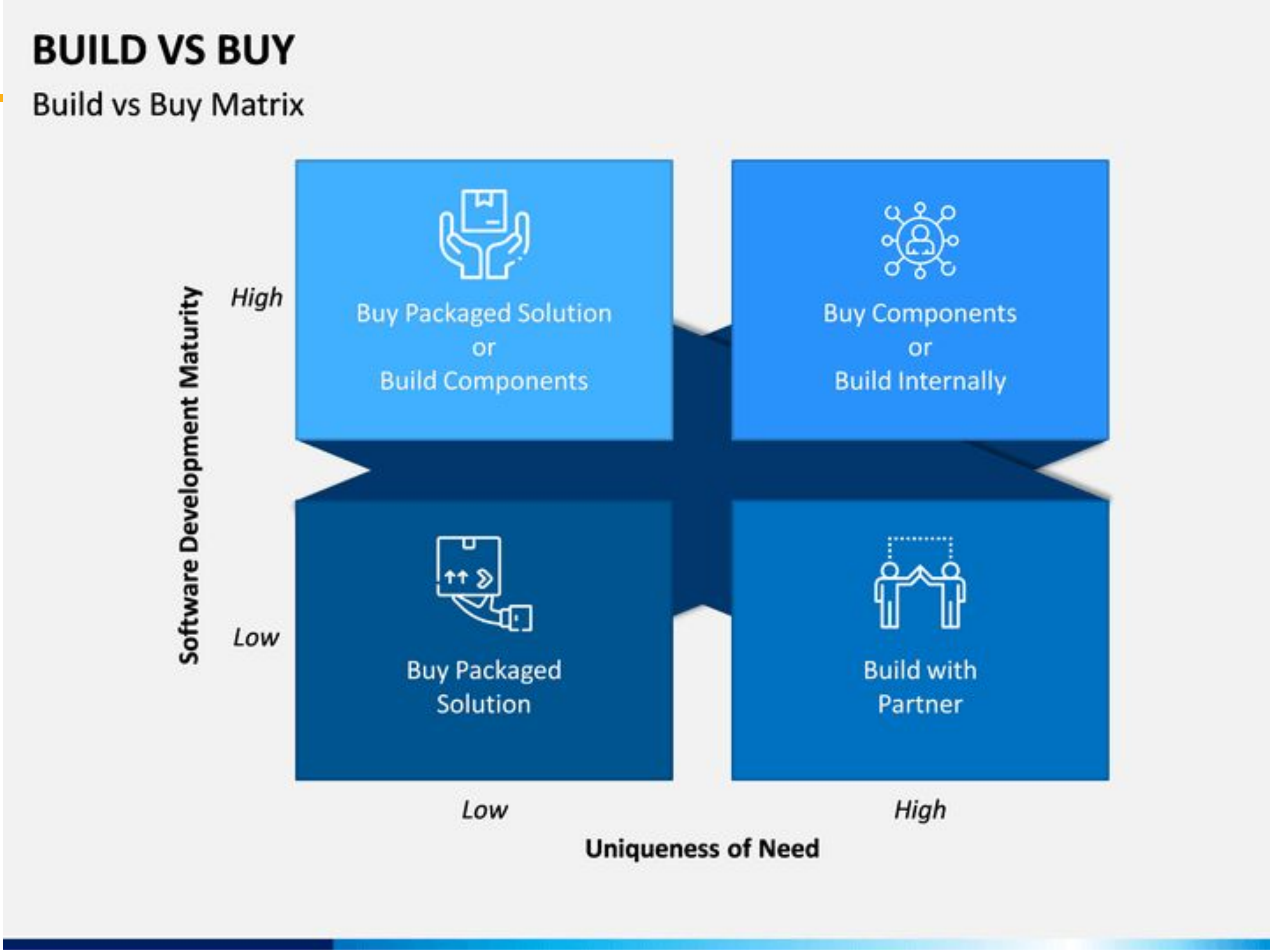
# Software Project Management

# Contact Session # 2

Prof. Narasimha Bolloju, BITS-Pilani, Hyderabad Campus

# Outline

- Buy vs. build decision
- Plan-driven and agile process models
  - Some popular models
  - Selection of right process model for a given project
- Agile paradigm shift
- Overview of Scrum and user stories



# BUILD VS BUY

## Build vs Buy Matrix

Software Development Maturity

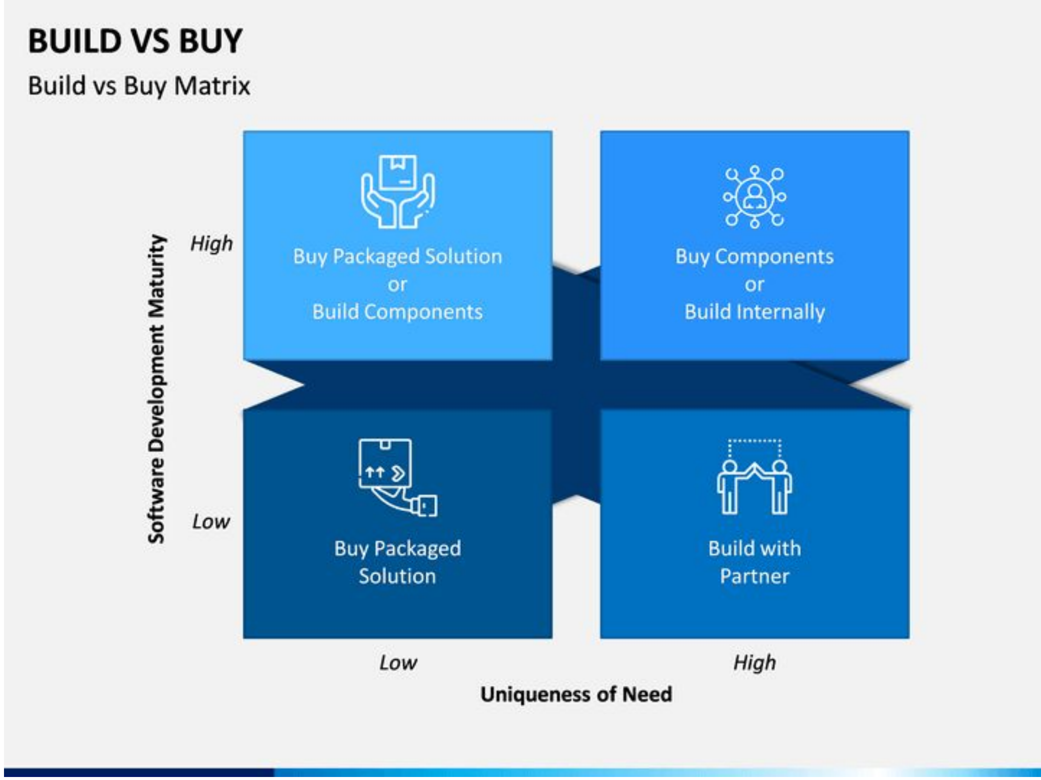
| | Low | High |
|---|---|---|
| High | Buy Packaged Solution<br>or<br>Build Components | Buy Components<br>or<br>Build Internally |
| Low | Buy Packaged Solution | Build with Partner |

Uniqueness of Need

build-vs-buy-slide3_1.png

# Activity CS2 #1

Which option is best suited for the Mux Core System(s) at ACL?  And, why?



**BUILD VS BUY**

Build vs Buy Matrix

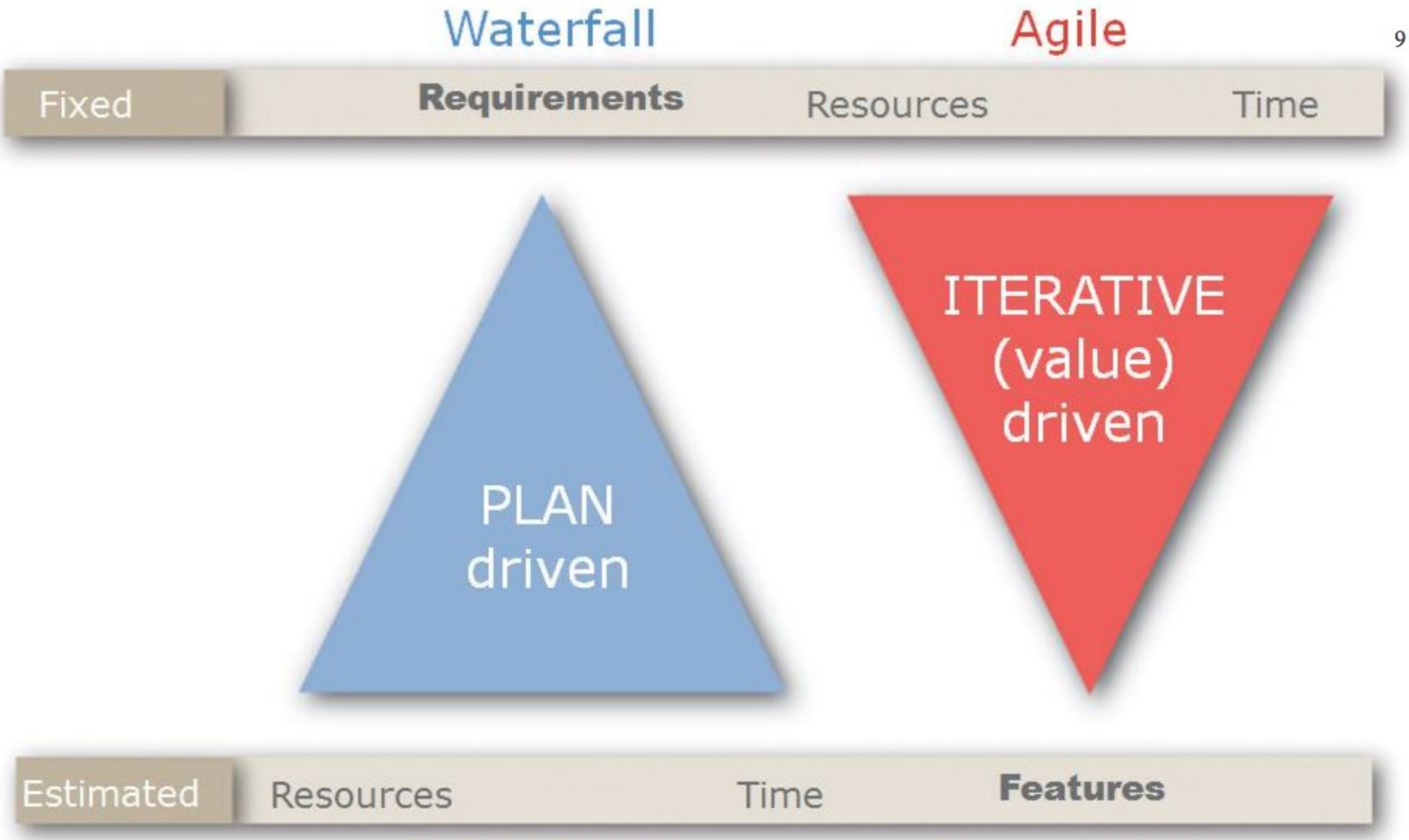
| Software Development Maturity | Low (Uniqueness of Need) | High (Uniqueness of Need) |
|---|---|---|
| High | Buy Packaged Solution or Build Components | Buy Components or Build Internally |
| Low | Buy Packaged Solution | Build with Partner |

# Plan-driven vs. agile process models

- In plan-driven process models, activities are planned in advance and progress is measured against the plan
- In agile process models, planning is incremental and it is easier to change the process to reflect changing requirements
- There is no right or wrong process model; most organizations/teams include elements from both in their process models

# The Agile Paradigm Shift

SEER by GALORATH



Waterfall

Agile

Fixed **Requirements** Resources Time

PLAN driven

ITERATIVE (value) driven

Estimated Resources Time **Features**

The plan creates cost/schedule estimates

Release themes and feature intent drive estimates

# Specific elements of the paradigm shift

- Customer-driven vs. customer-centric
- Change as an asset vs a risk
- Internally vs externally managed
- Teamwork vs. individual performance

https://blog.planview.com/4-paradigm-shifts-that-traditional-project-management-teams-must-embrace-to-make-agile-work/

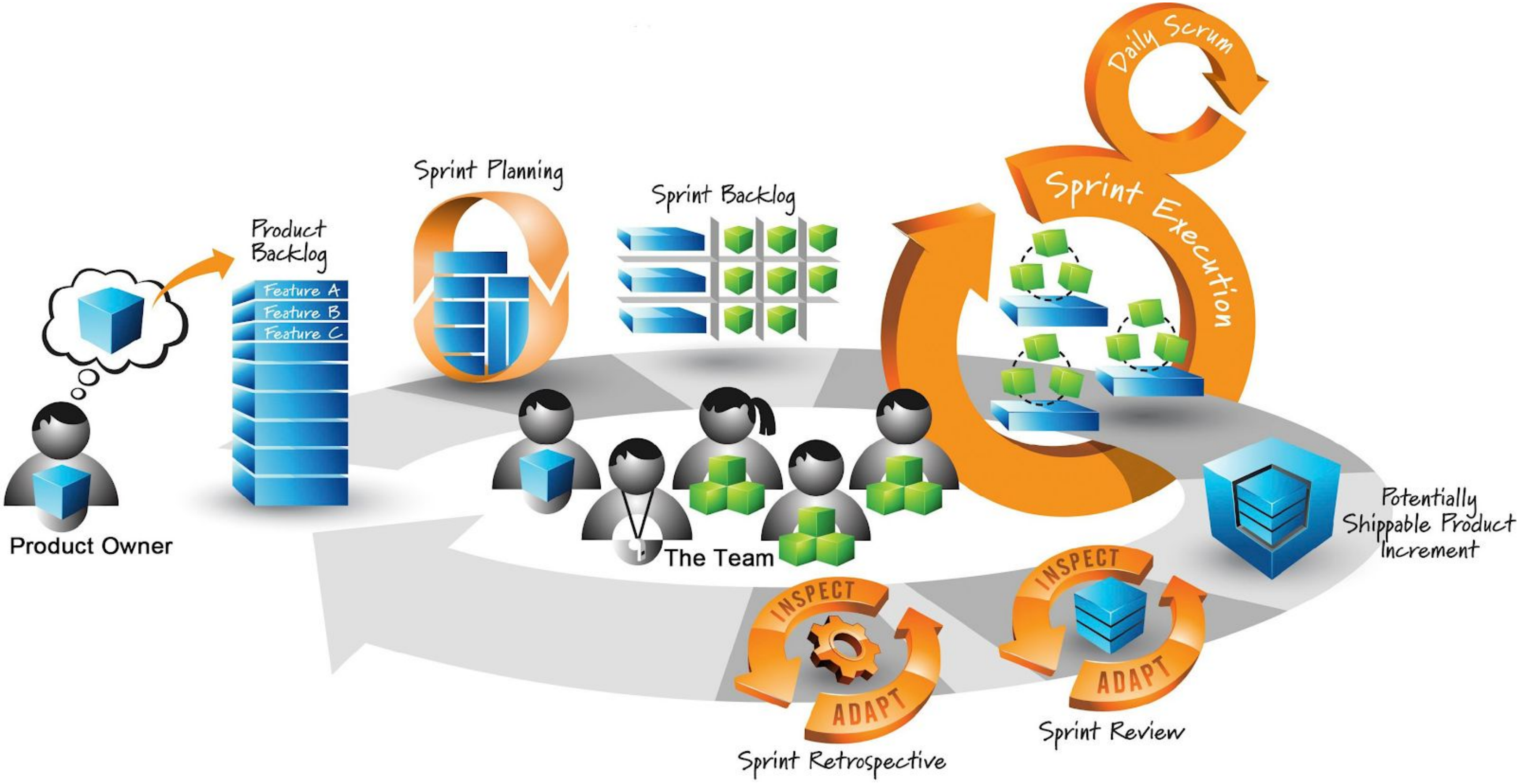
Also, read https://blog.planview.com/dos-and-donts-for-agile-team-success/

# Activity CS2 #2

Identify pros and cons of using the following process models for the Mux-Core Project

- waterfall
- iterative/incremental
- scrum (agile)

# Agile method Scrum - Overview



http://www.programmeronrails.com/2016/03/20/scrum-overview/

# Product Backlog written as User Stories

User Story Template

- As a user/role, I want to functionality/feature so that benefit

Example user stories in Library System

- As a student, I want to reserve books I want so that I can borrow the books as soon as they are available
- As a faculty, I want to recommend books for my courses so that my students will be able to read those books

Read https://www.mountaingoatsoftware.com/agile/user-stories

# Activity - CS02#4

Write four or five user stories for the Sales system of Mux-Core project at ACL

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

# SS ZG622:

# Software Project Management

# Contact Session # 3

Prof. Narasimha Bolloju, BITS-Pilani, Hyderabad Campus

# Outline

- Agile process - Scrum sprint planning
  - Product backlog (evolving system scope)
  - **Prioritizing and sprint backlog**
  - **Estimating story points**
  - Burndown charts for project management
  - Decomposition of complex user stories

# Mux-Core Sales – Expected functionality

| As a/an | I want to | So that |
|---|---|---|
| Supervisor | view daily sales performance of movies on dashboard | I can provide my feedback to my manager |
| Manager | view sales performance of various movies and screens | I can analyze sales performance |
| Manager | compare sales of movies with those at other multiplexes | I can analyze differences and take necessary actions |
| Manager | set discounts on specific shows | the multiplex can attract more customers |
| Manager | receive notifications about sales deviations | I can review the sales pattern and take necessary actions |
| Counter Staff | sign in for a sales duty session | I can sell tickets to customers |
| Counter Staff | sell movie tickets to customers | they can be admitted to watch movies |
| Counter Staff | sign out of sales duty session | my session details can be recorded and closed |
| Customer | search for movies and shows | I can select a movie show I am interested in |
| Customer | register for a loyalty card | I can buy tickets online and rate movies |
| Customer | buy tickets for a selected movie show | I/we can be admitted to watch that movie show |
| Customer | rate and review movies I have seen | other customers can read from my rating/review |
| Customer | view my loyalty card points balance | I can purchase tickets with those points |

# Product backlog prioritization by “value”

| As a/an | I want to |
|---|---|
| Manager | view sales performance of various movies and screens |
| Manager | compare sales of movies with those at other multiplexes |
| Manager | set discounts on specific shows |
| Manager | receive notifications about sales deviations from targets |
| Counter Staff | sign in for a sales duty session |
| Counter Staff | sell movie tickets to customers |
| Counter Staff | sign out of sales duty session |
| Customer | search for movies and shows |
| Customer | register for a loyalty card |
| Customer | buy tickets for a selected movie show |
| Customer | rate and review movies I have seen |
| Customer | view my loyalty card points balance |

# Activity  CS3#1

Prioritize the given user stories by changing the order of the stories (high value ones at the top)

https://pollev.com/narsibolloju019

# Scrum - estimation

- Traditional software teams give estimates in a time format: days, weeks, months.
- Many agile teams, however, have transitioned to story points. Story points rate the relative effort of work using
  - Fibonacci-like format: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
  - Powers of 2: 1, 2, 4, 8, 16, 32, 64, …
  - T-shirt sizes: XXS, XS, S, M, L, XL, XXL
  - …

# Planning Poker – estimation technique

- All participants use numbered playing cards and estimate the items
- Individual estimates are done and discussion is raised when there are large differences
- This process is repeated till the whole team reached consensus about the agreed estimation (which is not an average of estimates!)

https://technology.amis.nl/2016/03/23/8-agile-estimation-techniques-beyond-planning-poker/

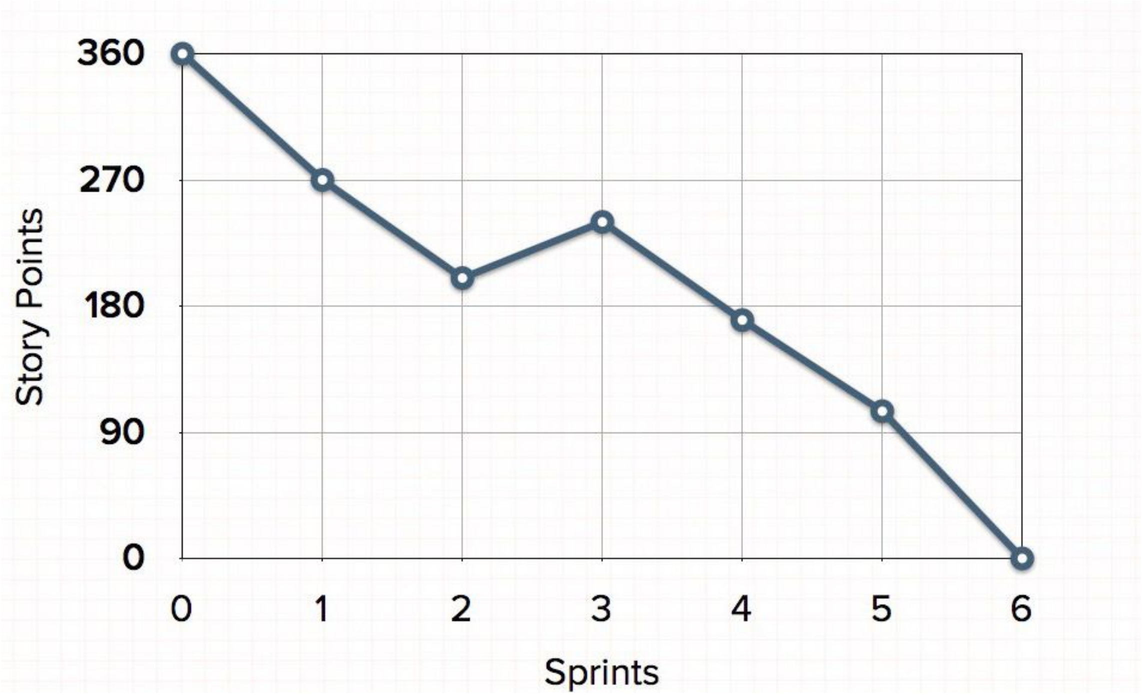
# Activity CS3#2

Estimation of “story points” for

“**set discounts on specific shows**”

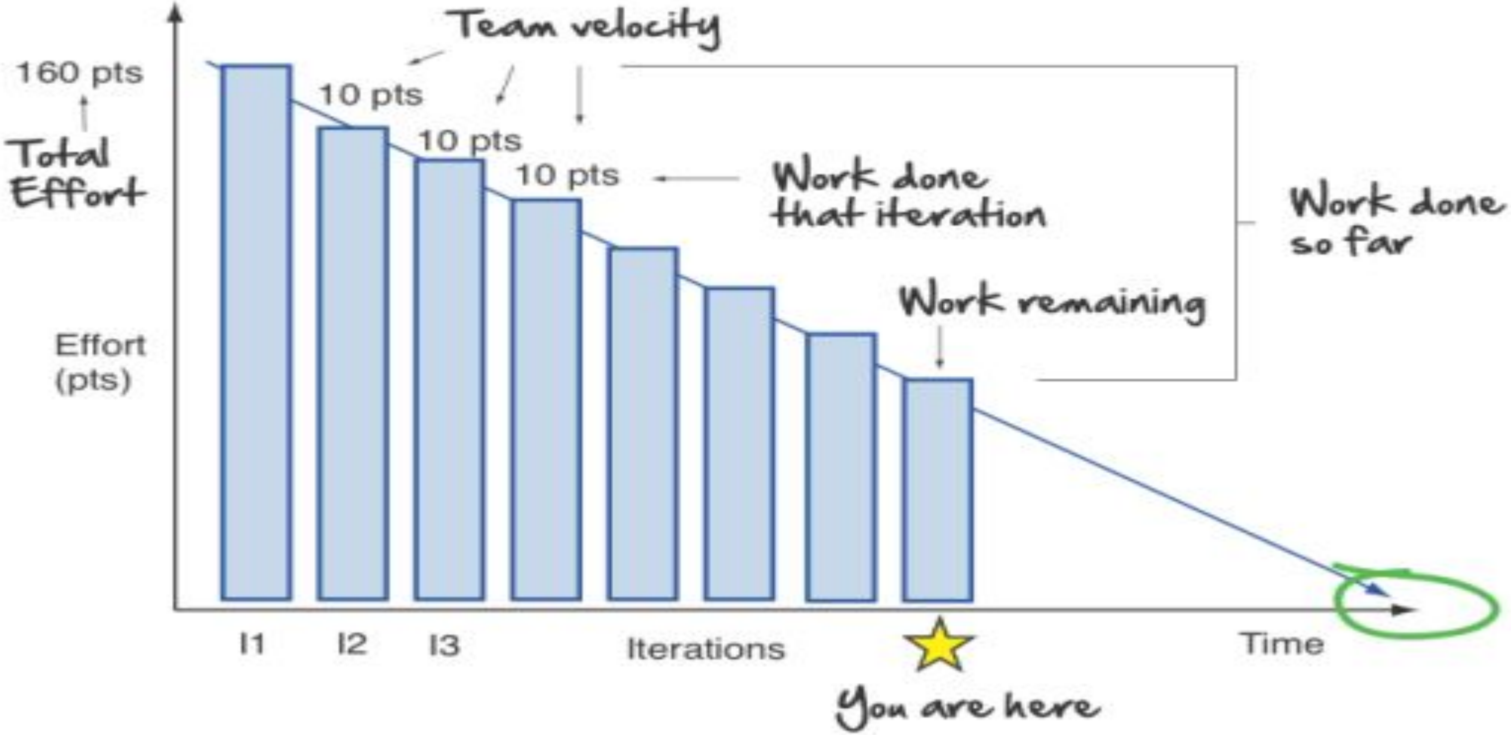
using powers of 2: 1, 2, 4, 8, 16, 32, …

# Burndown Chart - example



*

# Burndown Charts



# Activity CS3#3

Decompose the user story

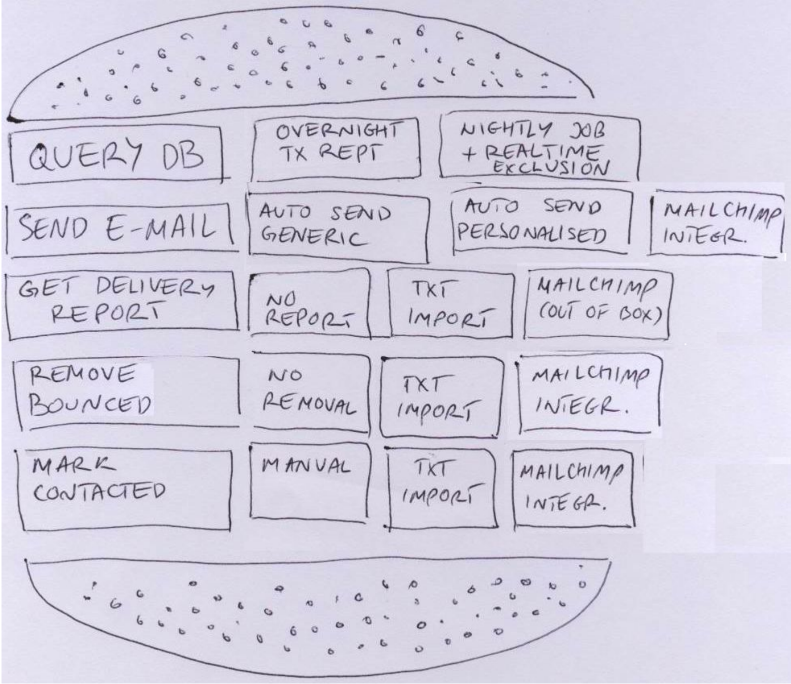
“**sell movie tickets to customers**”

into several simpler user stories

# User story decomposition - vertical slicing

https://gojko.net/2012/01/23/splitting-user-stories-the-hamburger-method/

# User story decomposition - vertical slicing



https://gojko.net/2012/01/23/splitting-user-stories-the-hamburger-method/

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

# SS ZG622:

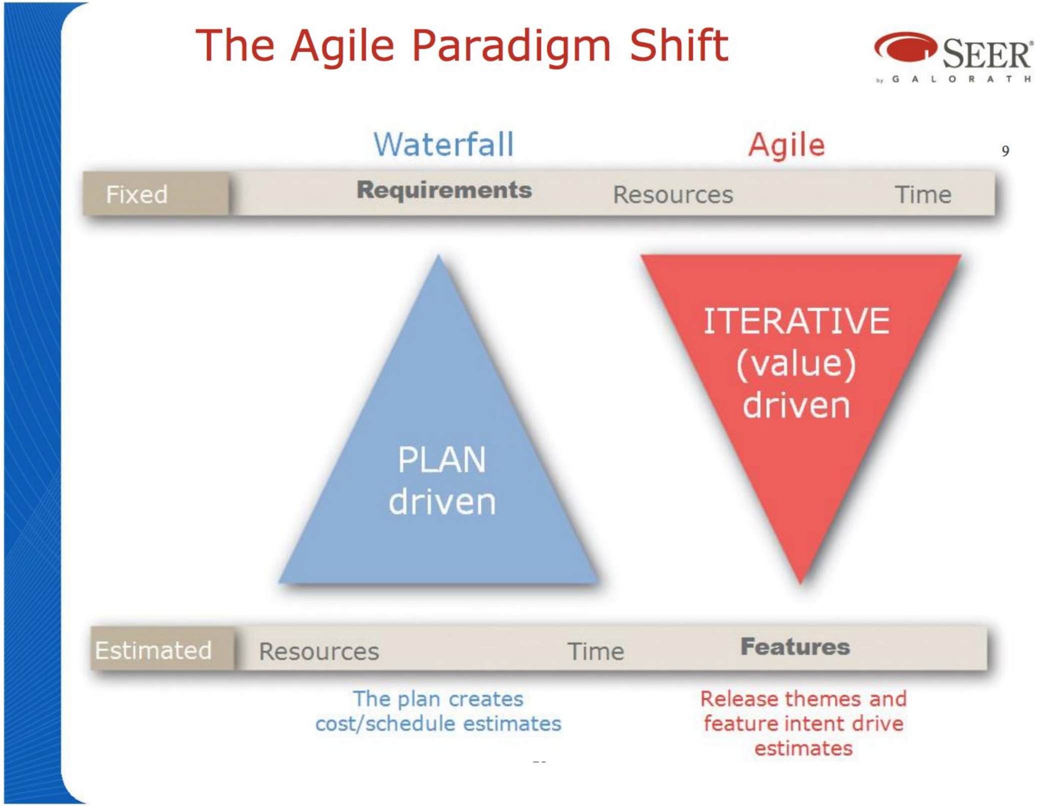
# Software Project Management

# Contact Session # 5

Prof. Narasimha Bolloju, BITS-Pilani, Hyderabad Campus

# Outline

- Effort estimation in plan-driven processes
- Function point estimation method
- Function points to effort
- Function points to SLOC



## The Agile Paradigm Shift

SEER by GALORATH

9

Waterfall

Agile

Fixed **Requirements** Resources Time

Estimated Resources Time **Features**

The plan creates cost/schedule estimates

Release themes and feature intent drive estimates

# Mux-Core Sales – Expected functionality

| As a/an | I want to | So that |
|---|---|---|
| Supervisor | view daily sales performance of movies on dashboard | I can provide my feedback to my manager |
| Manager | view sales performance of various movies and screens | I can analyze sales performance |
| Manager | compare sales of movies with those at other multiplexes | I can analyze differences and take necessary actions |
| Manager | set discounts on specific shows | the multiplex can attract more customers |
| Manager | receive notifications about sales deviations | I can review the sales pattern and take necessary actions |
| Counter Staff | sign in for a sales duty session | I can sell tickets to customers |
| Counter Staff | sell movie tickets to customers | they can be admitted to watch movies |
| Counter Staff | sign out of sales duty session | my session details can be recorded and closed |
| Customer | search for movies and shows | I can select a movie show I am interested in |
| Customer | register for a loyalty card | I can buy tickets online and rate movies |
| Customer | buy tickets for a selected movie show | I/we can be admitted to watch that movie show |
| Customer | rate and review movies I have seen | other customers can read from my rating/review |
| Customer | view my loyalty card points balance | I can purchase tickets with those points |

# Activity CS5 #1

Give a rough estimate of
the total effort in person-months for developing
the **Mux-Core sales system** with the
requirements listed on the previous slide
(pollev.com/narsibolloju019)

# Overall effort estimation

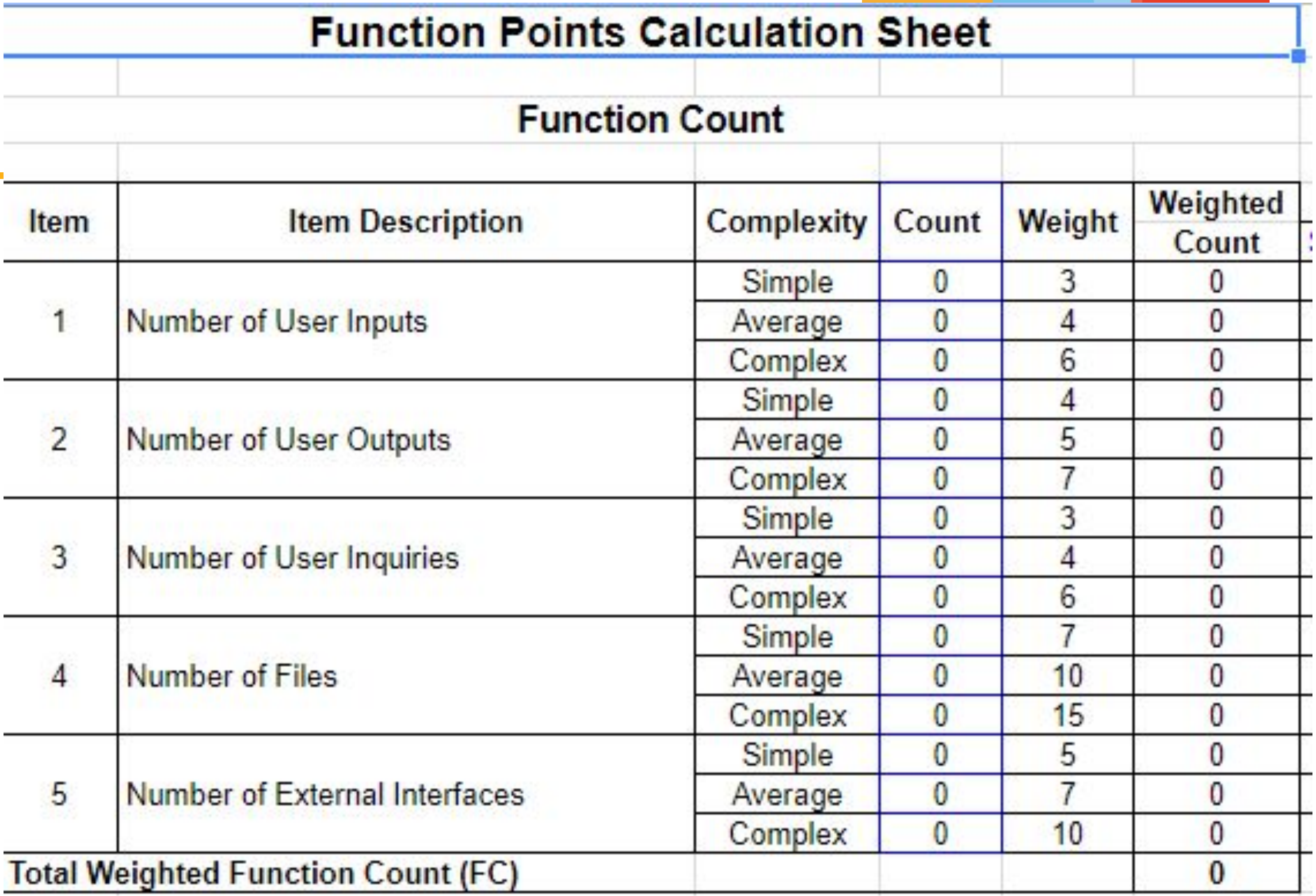
- Based on specifications of a software application
- Typical approaches: function point method, SLOC/KLOC method, object points method, OR purely experience-based
- Combines programmer productivity (historical data), development and implementation technologies, and other application characteristics
- Often the effort is represented in terms of person-months

# Activity CS5 #2

Give a rough estimate of
the total effort in person-months for developing
the **Mux-Core sales system** considering
individual functions/user stories

Use this Google Sheet first and then submit your
estimate in Chat

# Function point estimation

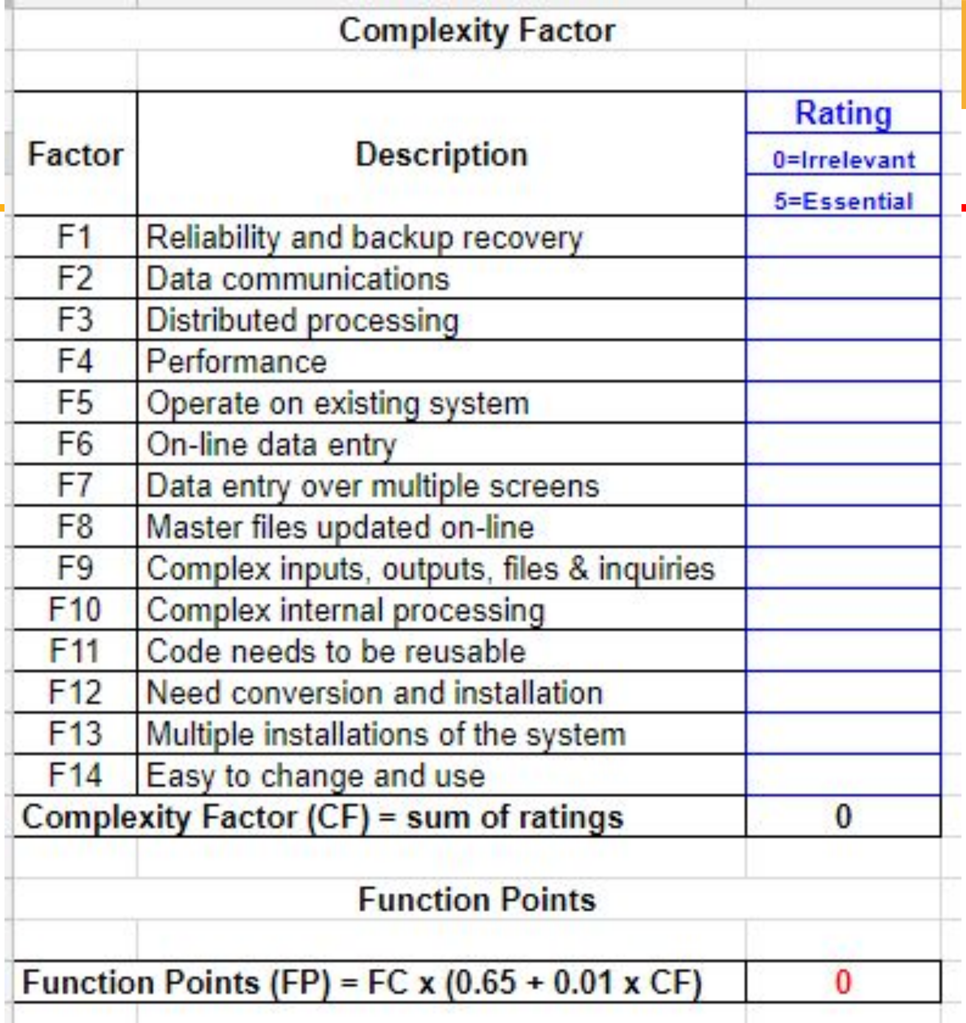


## Function Points Calculation Sheet

### Function Count

| Item | Item Description | Complexity | Count | Weight | Weighted Count |
|---|---|---|---|---|---|
| 1 | Number of User Inputs | Simple | 0 | 3 | 0 |
| | | Average | 0 | 4 | 0 |
| | | Complex | 0 | 6 | 0 |
| 2 | Number of User Outputs | Simple | 0 | 4 | 0 |
| | | Average | 0 | 5 | 0 |
| | | Complex | 0 | 7 | 0 |
| 3 | Number of User Inquiries | Simple | 0 | 3 | 0 |
| | | Average | 0 | 4 | 0 |
| | | Complex | 0 | 6 | 0 |
| 4 | Number of Files | Simple | 0 | 7 | 0 |
| | | Average | 0 | 10 | 0 |
| | | Complex | 0 | 15 | 0 |
| 5 | Number of External Interfaces | Simple | 0 | 5 | 0 |
| | | Average | 0 | 7 | 0 |
| | | Complex | 0 | 10 | 0 |
| Total Weighted Function Count (FC) | | | | | 0 |

# Function point estimation



Complexity Factor

| Factor | Description | Rating<br>0=Irrelevant<br>5=Essential |
|---|---|---|
| F1 | Reliability and backup recovery | |
| F2 | Data communications | |
| F3 | Distributed processing | |
| F4 | Performance | |
| F5 | Operate on existing system | |
| F6 | On-line data entry | |
| F7 | Data entry over multiple screens | |
| F8 | Master files updated on-line | |
| F9 | Complex inputs, outputs, files & inquiries | |
| F10 | Complex internal processing | |
| F11 | Code needs to be reusable | |
| F12 | Need conversion and installation | |
| F13 | Multiple installations of the system | |
| F14 | Easy to change and use | |
| Complexity Factor (CF) = sum of ratings | | 0 |

Function Points

| | |
|---|---|
| Function Points (FP) = FC x (0.65 + 0.01 x CF) | 0 |

# Function point estimation - Complexity example

| Number of **views contained** | Number and source of **data tables** | | |
|---|---|---|---|
| | Total < 4<br>(<2 servers;<br><3 clients) | Total < 8<br>(<3 servers;<br>3 to 5 clients) | Total > 7<br>( >3 servers<br>> 5 clients) |
| <3 | simple | simple | average |
| 3 to 7 | simple | average | complex |
| >7 | average | complex | complex |

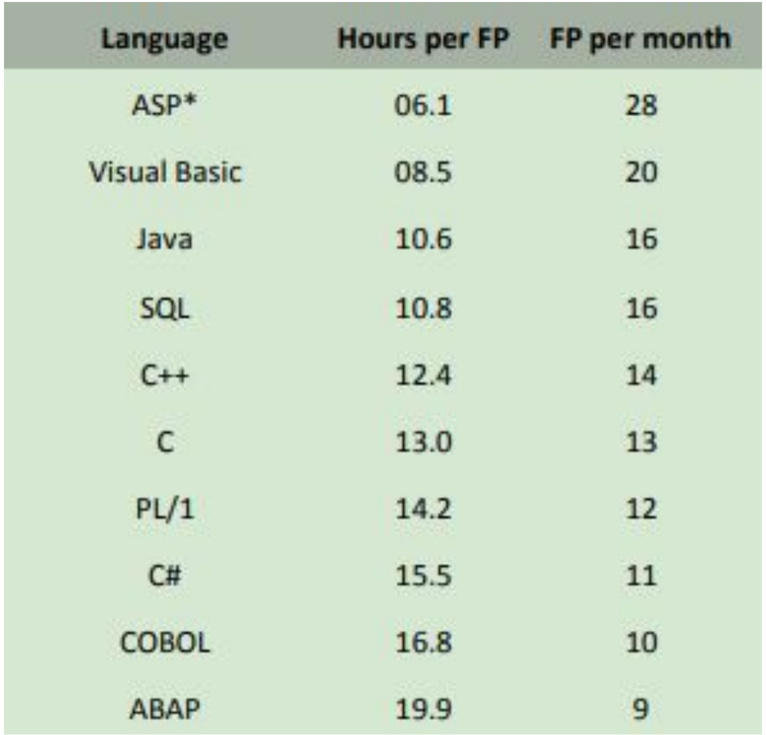
https://www.gristprojectmanagement.us/software-2/albrecht-function-point-analysis.html
https://www.gristprojectmanagement.us/software/albrecht-function-point-analysis.html

# CS5 #3 - FP estimation

**Estimate the function points for the specified functionality of Mux-Core system**

Instructions:
- Make a copy of this document "CS4 estimating function points"
- Estimate function points for F2 to F6 (for F1 it is already estimated)
- Post the estimated function points into the chat window

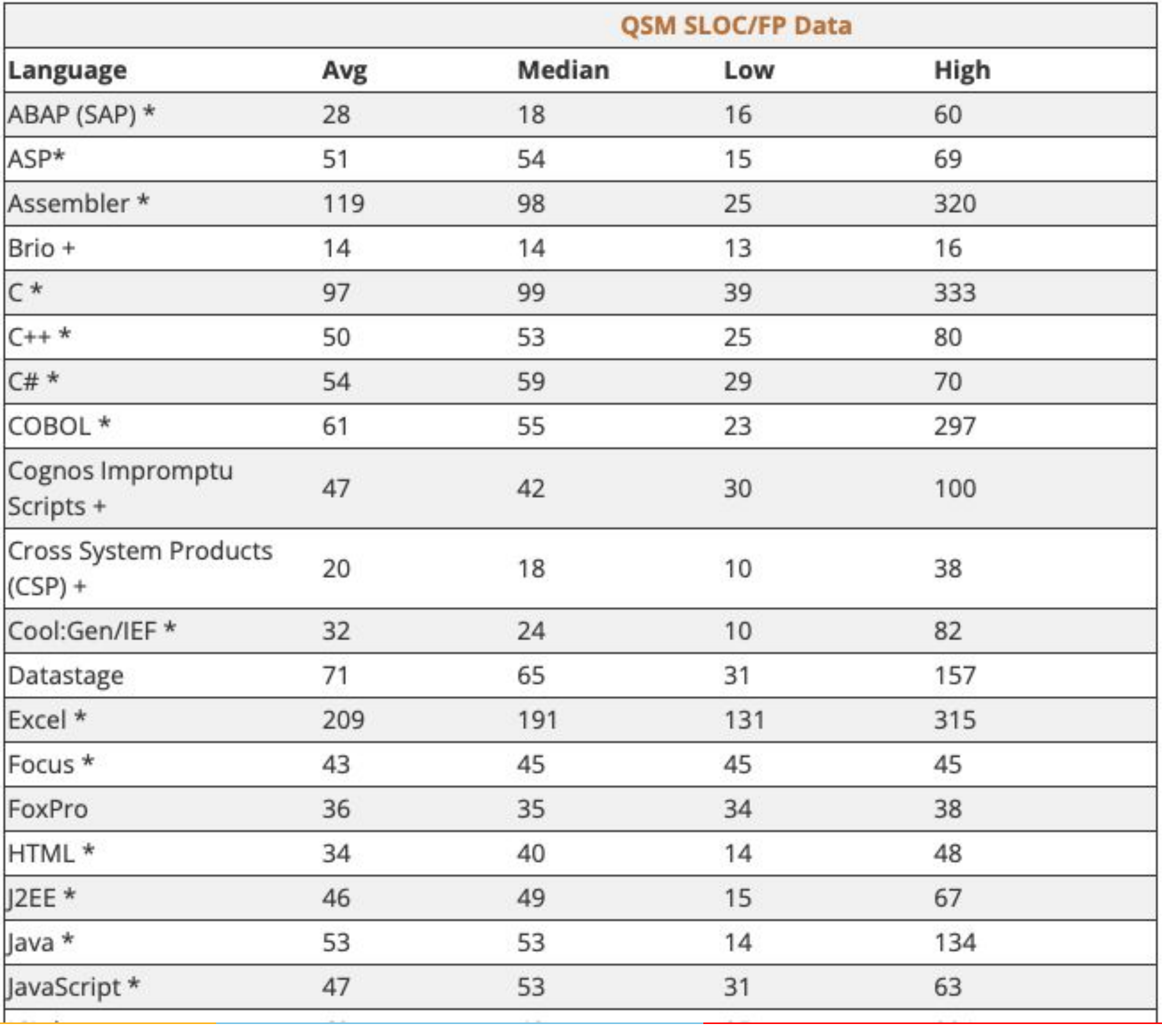
# FP productivity based on programming language (only an example)



| Language | Hours per FP | FP per month |
|---|---|---|
| ASP* | 06.1 | 28 |
| Visual Basic | 08.5 | 20 |
| Java | 10.6 | 16 |
| SQL | 10.8 | 16 |
| C++ | 12.4 | 14 |
| C | 13.0 | 13 |
| PL/1 | 14.2 | 12 |
| C# | 15.5 | 11 |
| COBOL | 16.8 | 10 |
| ABAP | 19.9 | 9 |

http://www.webratio.com/website/documentation/Case_Study_Productivity_with_WebRatio.pdf

# SLOC for function point - programming languages



| QSM SLOC/FP Data | | | | |
|---|---|---|---|---|
| Language | Avg | Median | Low | High |
| ABAP (SAP) * | 28 | 18 | 16 | 60 |
| ASP* | 51 | 54 | 15 | 69 |
| Assembler * | 119 | 98 | 25 | 320 |
| Brio + | 14 | 14 | 13 | 16 |
| C * | 97 | 99 | 39 | 333 |
| C++ * | 50 | 53 | 25 | 80 |
| C# * | 54 | 59 | 29 | 70 |
| COBOL * | 61 | 55 | 23 | 297 |
| Cognos Impromptu Scripts + | 47 | 42 | 30 | 100 |
| Cross System Products (CSP) + | 20 | 18 | 10 | 38 |
| Cool:Gen/IEF * | 32 | 24 | 10 | 82 |
| Datastage | 71 | 65 | 31 | 157 |
| Excel * | 209 | 191 | 131 | 315 |
| Focus * | 43 | 45 | 45 | 45 |
| FoxPro | 36 | 35 | 34 | 38 |
| HTML * | 34 | 40 | 14 | 48 |
| J2EE * | 46 | 49 | 15 | 67 |
| Java * | 53 | 53 | 14 | 134 |
| JavaScript * | 47 | 53 | 31 | 63 |

https://www.qsm.com/resources/function-point-languages-table

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

# SS ZG622:

# Software Project Management

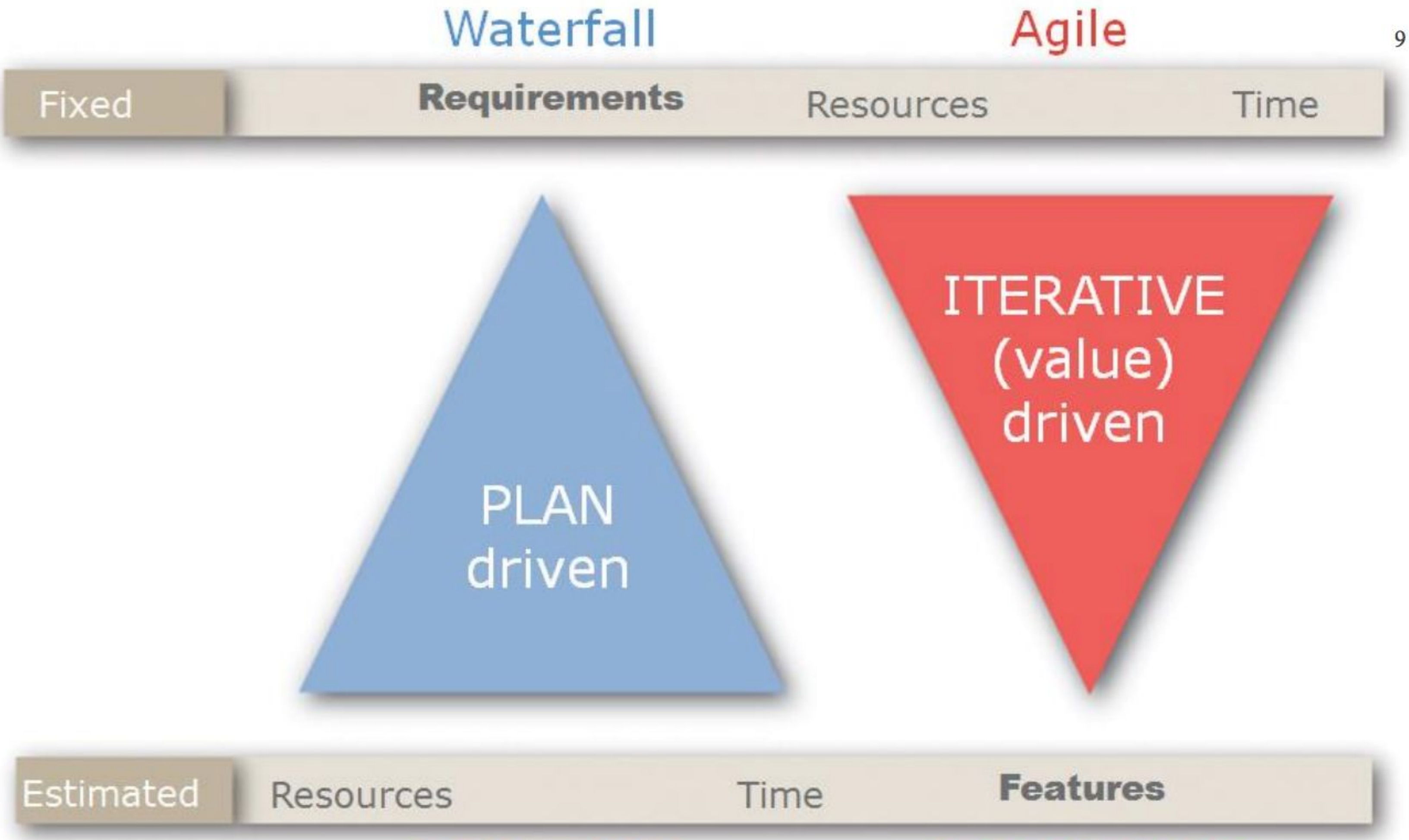
# Contact Session # 6

Prof. Narasimha Bolloju, BITS-Pilani, Hyderabad Campus

# Project planning - Topics

- Project planning in plan-driven methods
  - Mapping effort to duration (and schedule)
  - Example of effort distribution in waterfall methods
  - COCOMO II – overview and application
  - Work breakdown structure (WBS)
  - Gantt charts and critical path(s)
- Project planning in agile methods
  - Example of effort distribution in agile methods
  - Example tool - Pivotal Tracker (user stories, bugs, chores and milestones)

# The Agile Paradigm Shift



Waterfall

Agile

Fixed

**Requirements**

Resources

Time

PLAN driven

ITERATIVE (value) driven

Estimated

Resources

Time

**Features**

The plan creates cost/schedule estimates

Release themes and feature intent drive estimates

# Mapping effort to duration

- Effort in function points or KLOC (source lines of delivered code in thousands)
- Effort to duration
  - Experience-based and/or heuristics-based methods
  - Parametric methods like COCOMO81 and COCOMO II

# CS6-#1 Estimate project duration

Assuming that the effort and productivity for the Mux-core project are as follows:

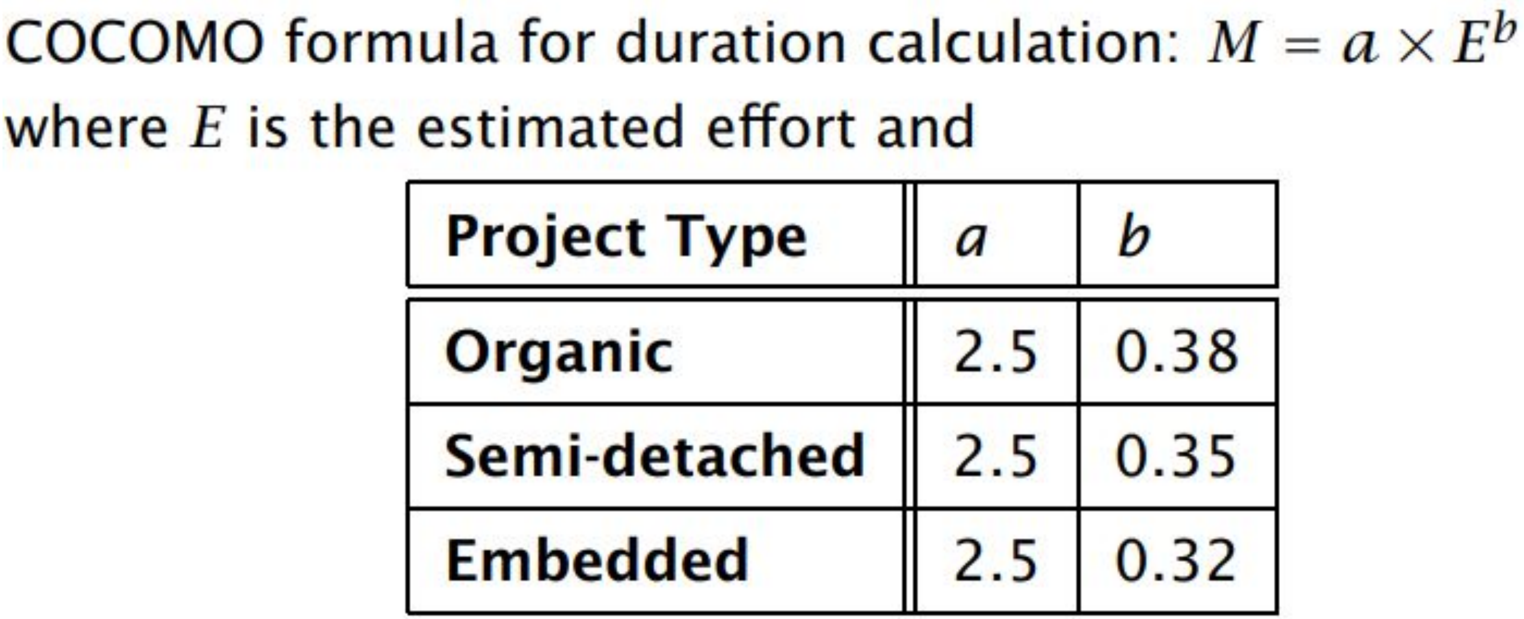
-Function points ~ 500

-Average productivity 10 function points per month per member

What should be the ideal project duration?

https://www.ifpug.org/wp-content/uploads/2017/04/IYSM.-Thirty-years-of-IFPUG.-Software-Economics-and-Function-Point-Metrics-Capers-Jones.pdf

# COCOMO formula for duration



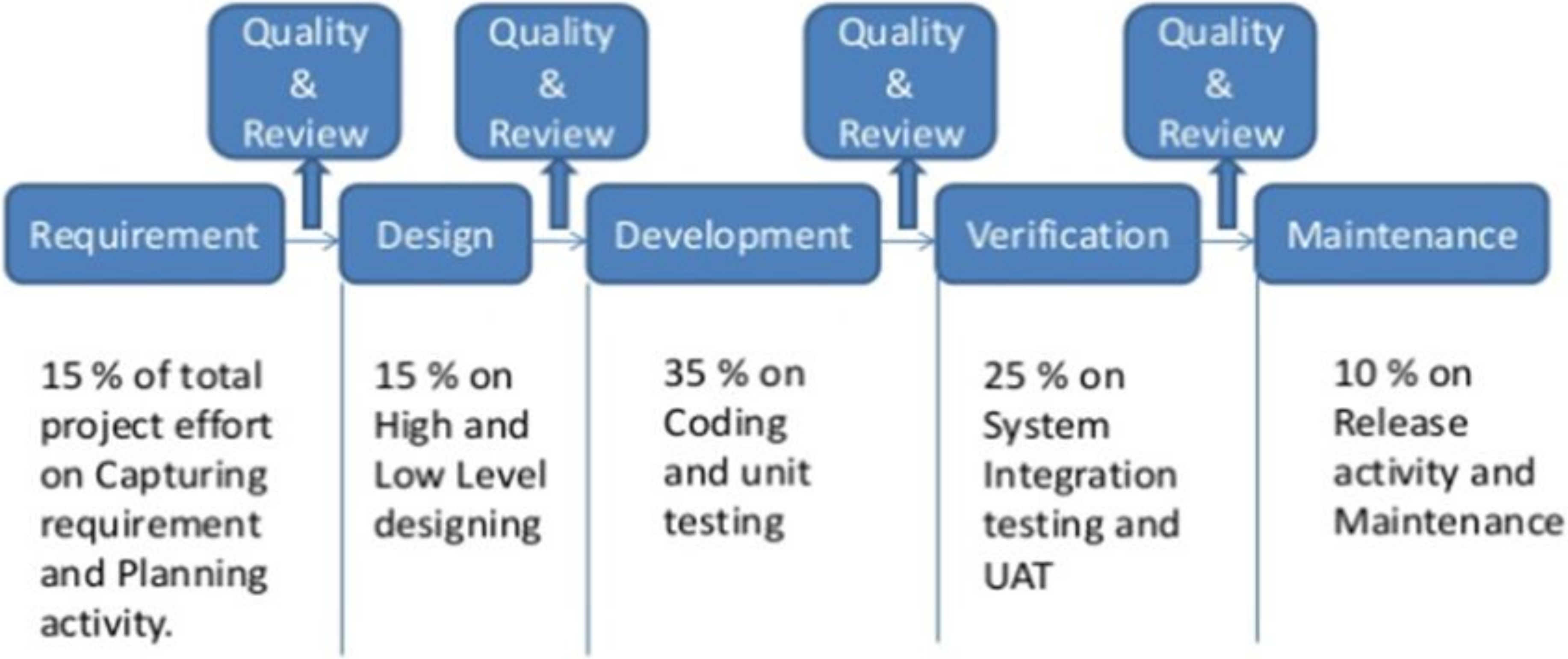
COCOMO formula for duration calculation: $M = a \times E^b$

where $E$ is the estimated effort and

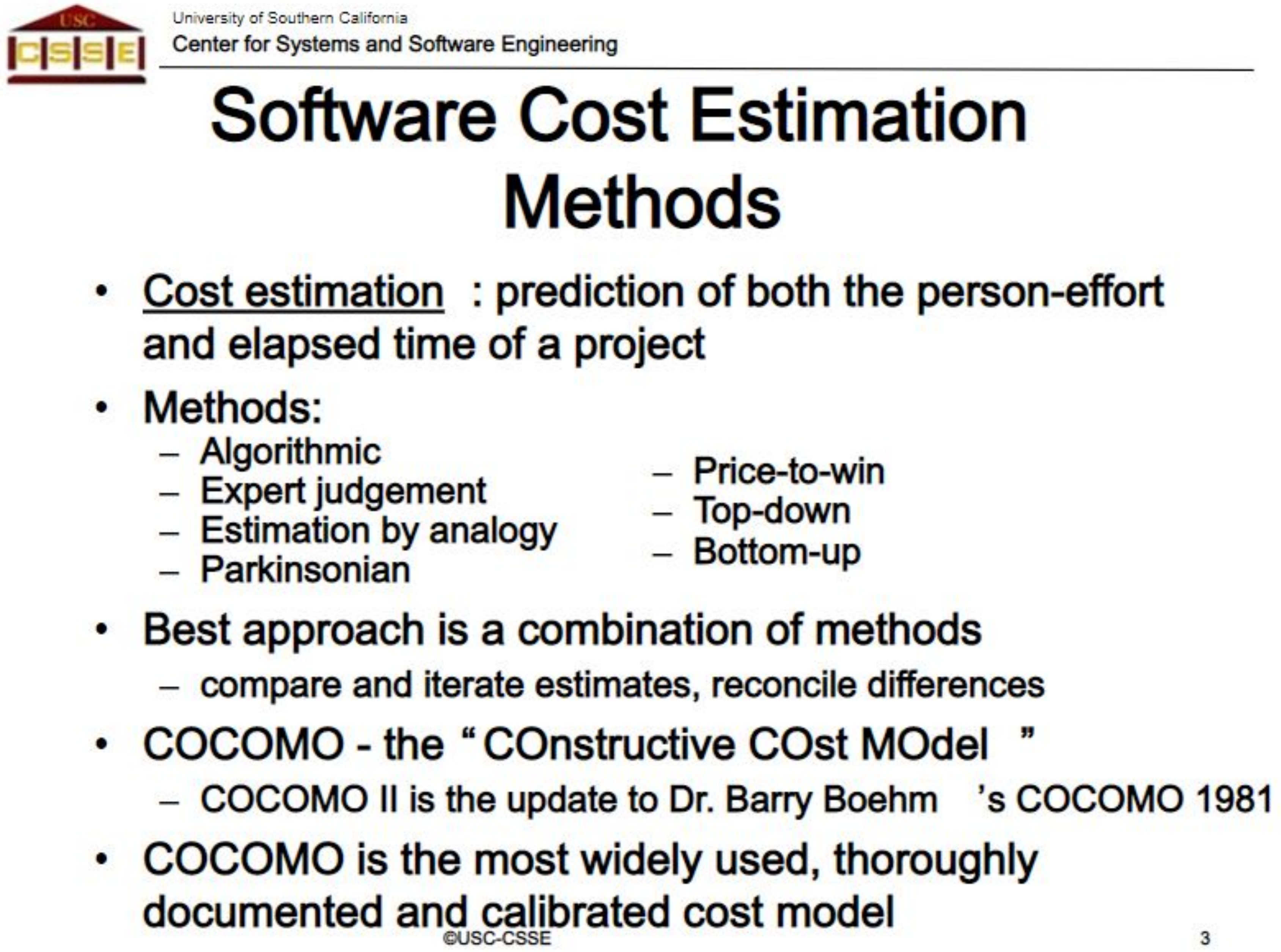
| Project Type | $a$ | $b$ |
|---|---|---|
| Organic | 2.5 | 0.38 |
| Semi-detached | 2.5 | 0.35 |
| Embedded | 2.5 | 0.32 |

Assuming that Mux-Core is a semi-detached project type with 500 fp, the project duration should be 22 months

# Example of effort distribution in waterfall process


Quality & Review
Quality & Review
Quality & Review
Quality & Review
Requirement
Design
Development
Verification
Maintenance
15 % of total project effort on Capturing requirement and Planning activity.
15 % on High and Low Level designing
35 % on Coding and unit testing
25 % on System Integration testing and UAT
10 % on Release activity and Maintenance

https://www.slideshare.net/brickedestimation/effort-distributiononwaterfallandagile



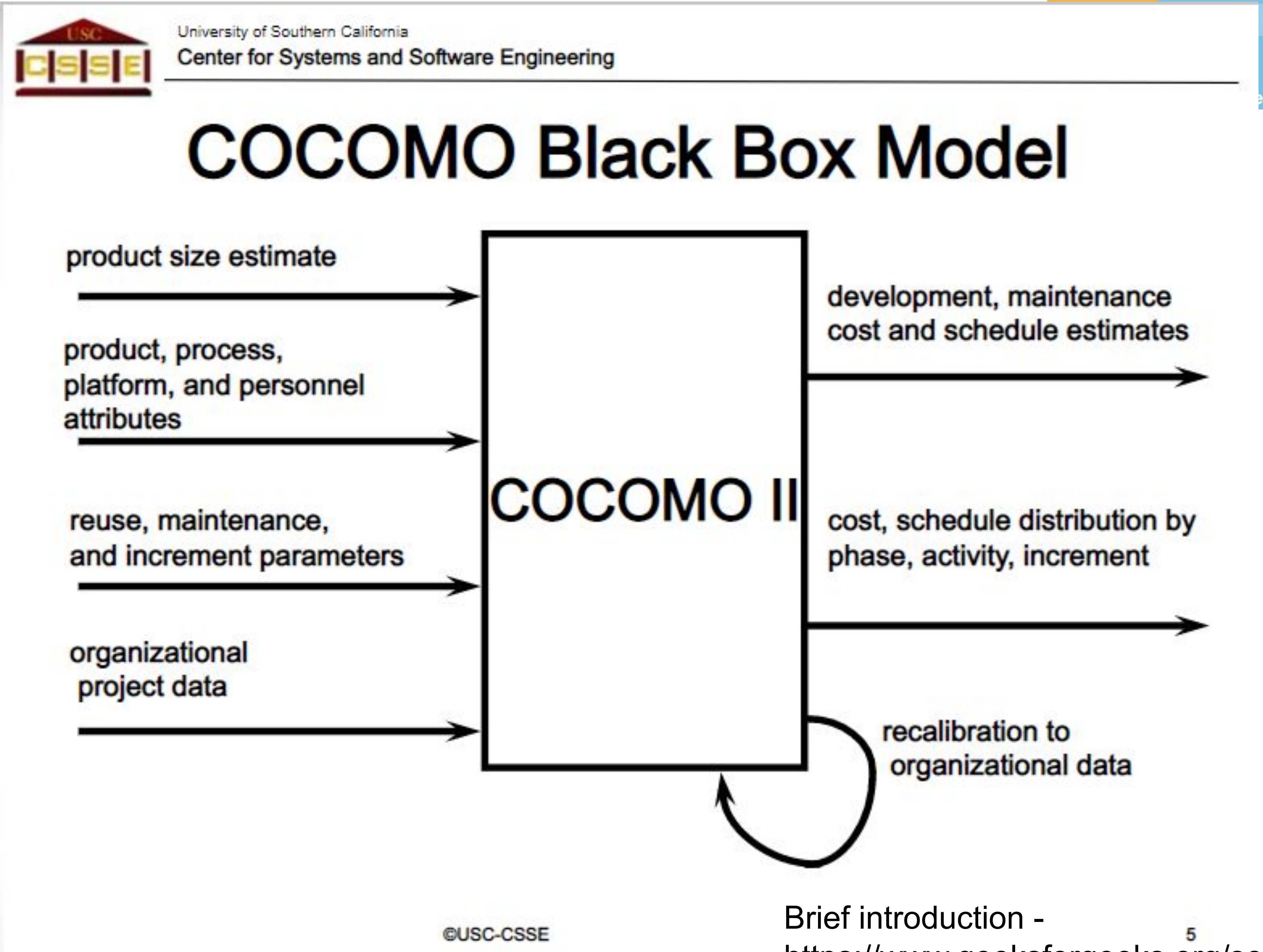
University of Southern California
Center for Systems and Software Engineering

# Software Cost Estimation Methods

- Cost estimation : prediction of both the person-effort and elapsed time of a project
- Methods:
  - Algorithmic
  - Expert judgement
  - Estimation by analogy
  - Parkinsonian
  - Price-to-win
  - Top-down
  - Bottom-up
- Best approach is a combination of methods
  - compare and iterate estimates, reconcile differences
- COCOMO - the “COnstructive COst MOdel ”
  - COCOMO II is the update to Dr. Barry Boehm ’s COCOMO 1981
- COCOMO is the most widely used, thoroughly documented and calibrated cost model

©USC-CSSE 3

https://slideplayer.com/slide/7947860/



University of Southern California
Center for Systems and Software Engineering

# COCOMO Black Box Model

product size estimate
product, process, platform, and personnel attributes
reuse, maintenance, and increment parameters
organizational project data
COCOMO II
development, maintenance cost and schedule estimates
cost, schedule distribution by phase, activity, increment
recalibration to organizational data

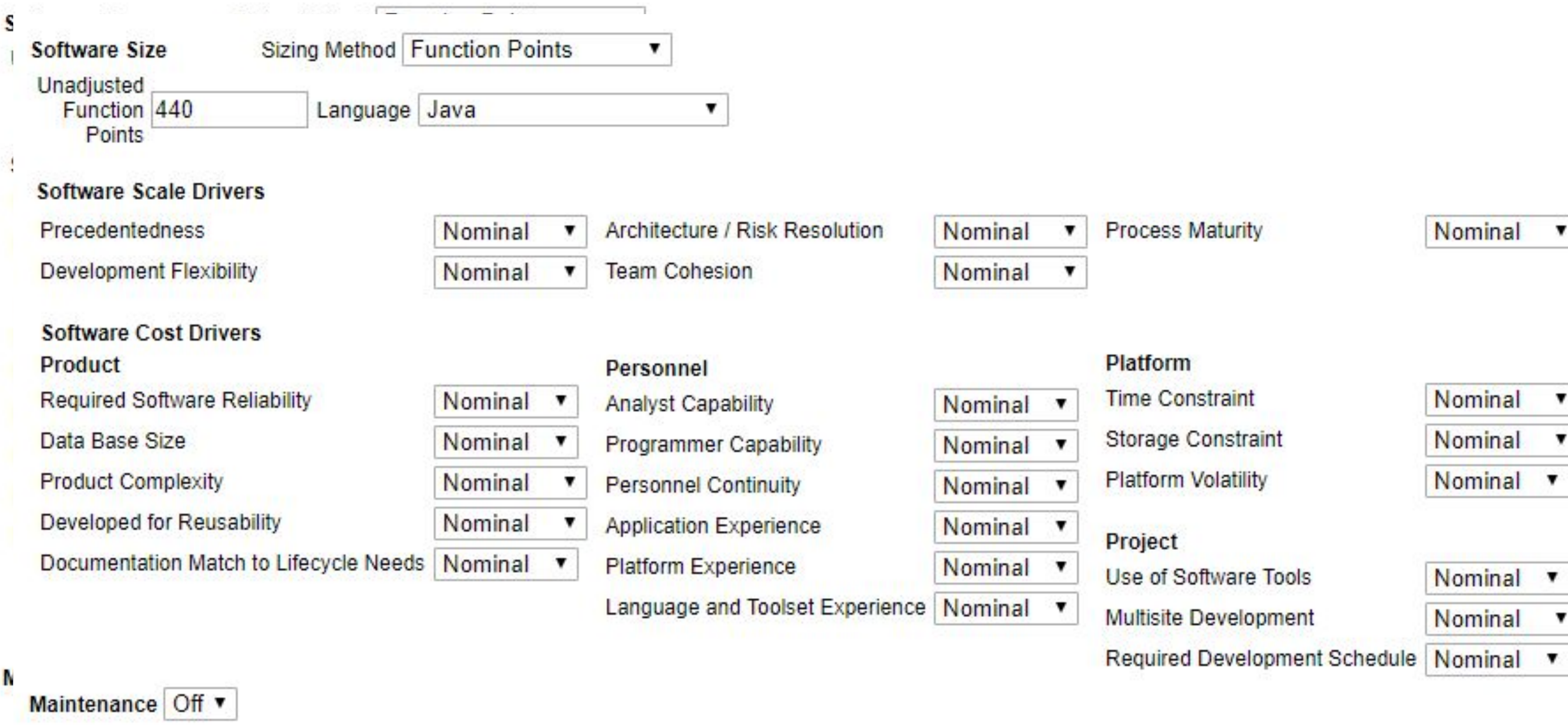
©USC-CSSE

5

https://slideplayer.com/slide/7947860/

Brief introduction - https://www.geeksforgeeks.org/software-engineering-cocomo-ii-model/

# COCOMO II Calculator (read chap 5 of the text book for details)



**Software Size**

Sizing Method: Function Points ▼

Unadjusted Function Points: 440

Language: Java ▼

**Software Scale Drivers**

| | | | | | |
|---|---|---|---|---|---|
| Precedentedness | Nominal ▼ | Architecture / Risk Resolution | Nominal ▼ | Process Maturity | Nominal ▼ |
| Development Flexibility | Nominal ▼ | Team Cohesion | Nominal ▼ | | |

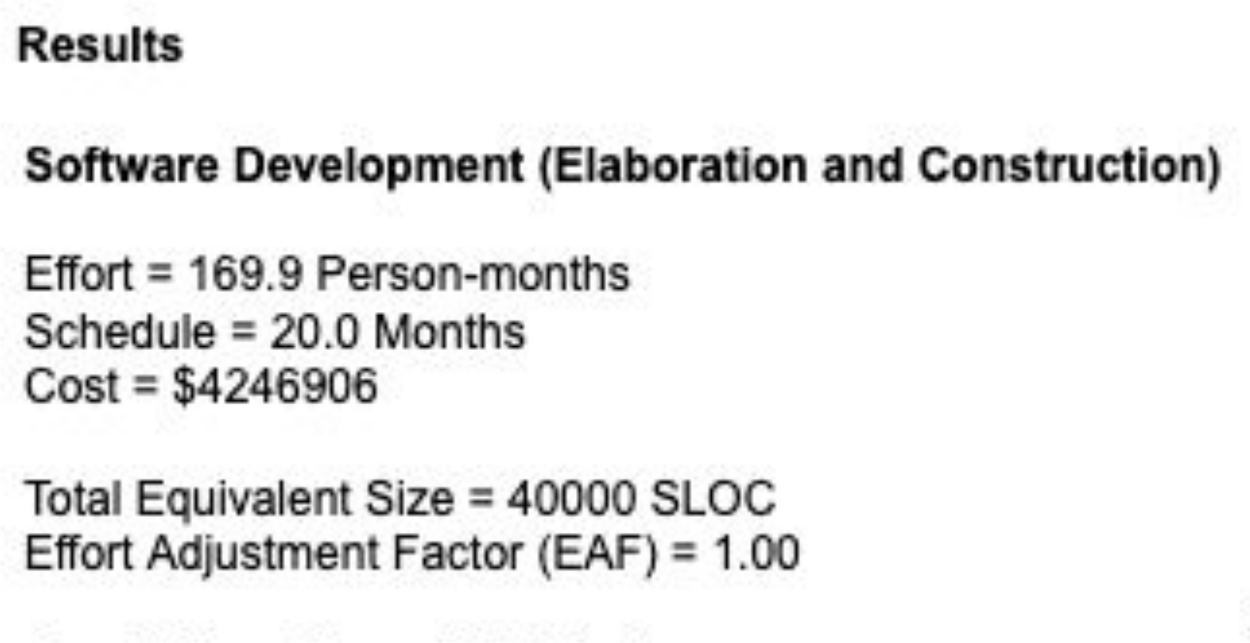
**Software Cost Drivers**

| **Product** | | **Personnel** | | **Platform** | |
|---|---|---|---|---|---|
| Required Software Reliability | Nominal ▼ | Analyst Capability | Nominal ▼ | Time Constraint | Nominal ▼ |
| Data Base Size | Nominal ▼ | Programmer Capability | Nominal ▼ | Storage Constraint | Nominal ▼ |
| Product Complexity | Nominal ▼ | Personnel Continuity | Nominal ▼ | Platform Volatility | Nominal ▼ |
| Developed for Reusability | Nominal ▼ | Application Experience | Nominal ▼ | **Project** | |
| Documentation Match to Lifecycle Needs | Nominal ▼ | Platform Experience | Nominal ▼ | Use of Software Tools | Nominal ▼ |
| | | Language and Toolset Experience | Nominal ▼ | Multisite Development | Nominal ▼ |
| | | | | Required Development Schedule | Nominal ▼ |

**Maintenance** Off ▼

http://softwarecost.org/tools/COCOMO/

# COCOMO II Calculator (read chap 5 of the text book for details)



**Results**

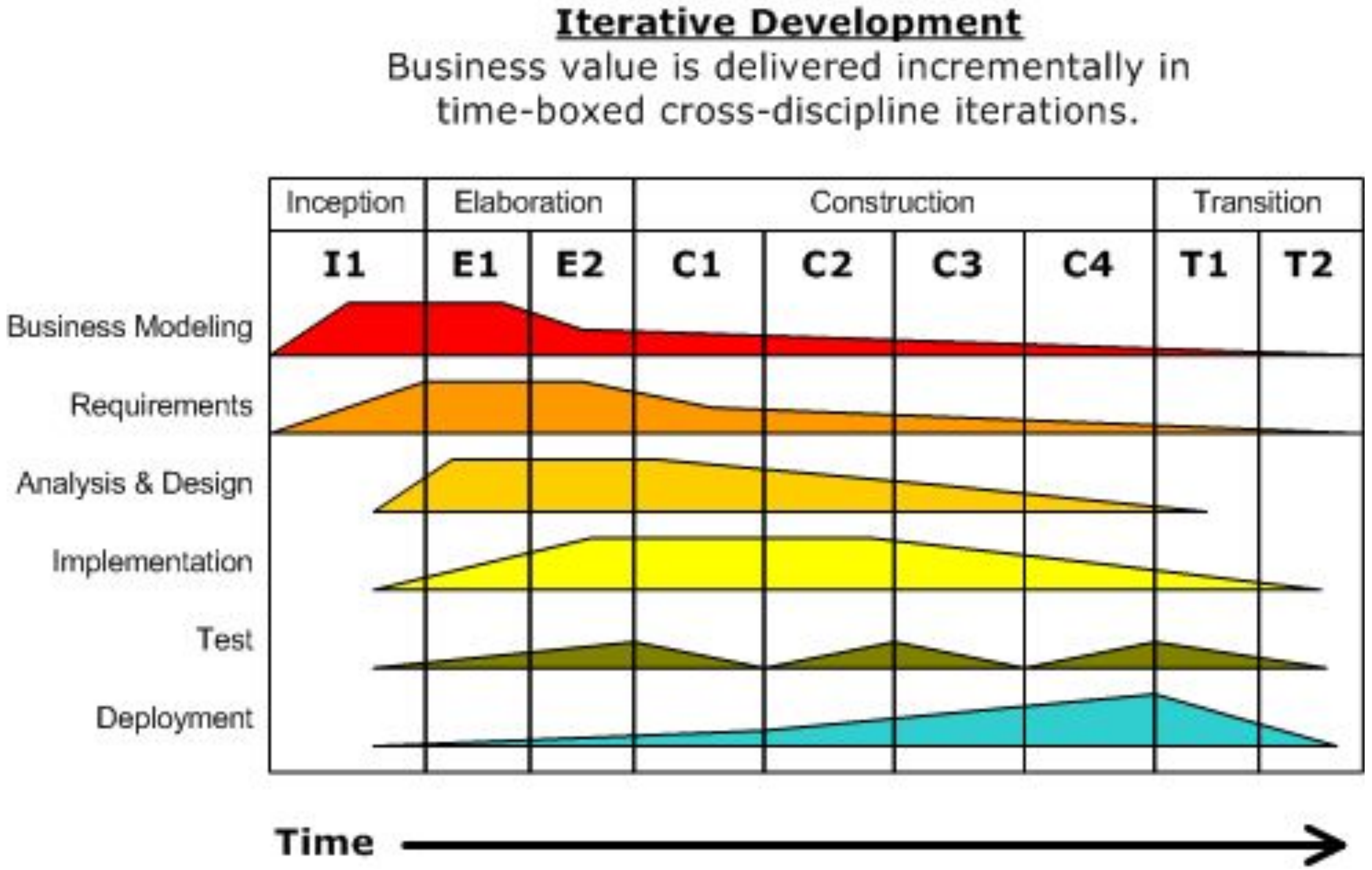
**Software Development (Elaboration and Construction)**

Effort = 169.9 Person-months
Schedule = 20.0 Months
Cost = $4246906

Total Equivalent Size = 40000 SLOC
Effort Adjustment Factor (EAF) = 1.00

Note: Assume $ = INR for interpretation of the inputs and results;; i.e., 25k rupees average salary per month

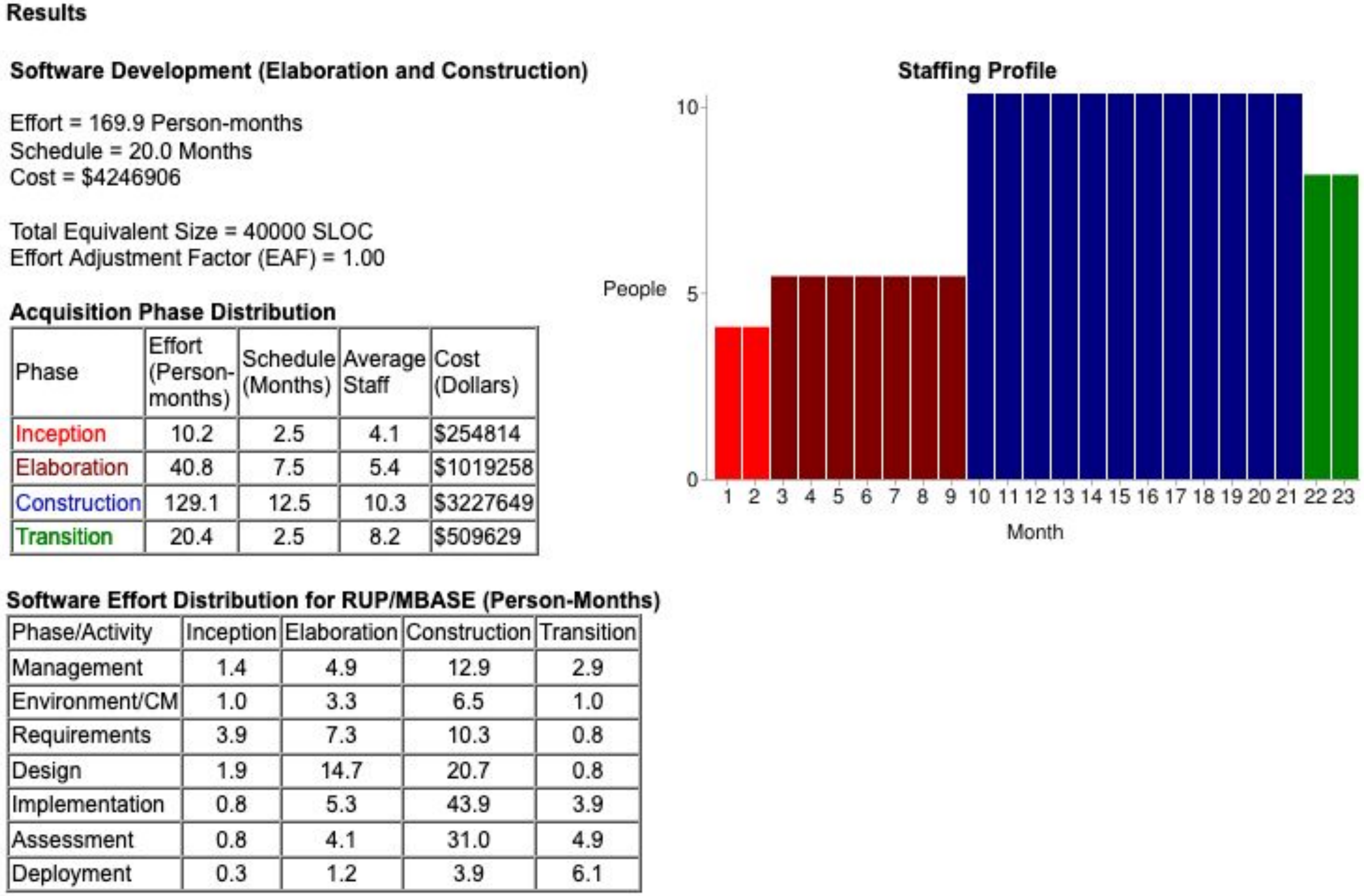
# Unified Process (for explaining estimation results)



**Iterative Development**

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

# COCOMO II Calculator



**Results**

**Software Development (Elaboration and Construction)**

Effort = 169.9 Person-months
Schedule = 20.0 Months
Cost = $4246906

Total Equivalent Size = 40000 SLOC
Effort Adjustment Factor (EAF) = 1.00

**Acquisition Phase Distribution**

| Phase | Effort (Person-months) | Schedule (Months) | Average Staff | Cost (Dollars) |
|---|---|---|---|---|
| Inception | 10.2 | 2.5 | 4.1 | $254814 |
| Elaboration | 40.8 | 7.5 | 5.4 | $1019258 |
| Construction | 129.1 | 12.5 | 10.3 | $3227649 |
| Transition | 20.4 | 2.5 | 8.2 | $509629 |

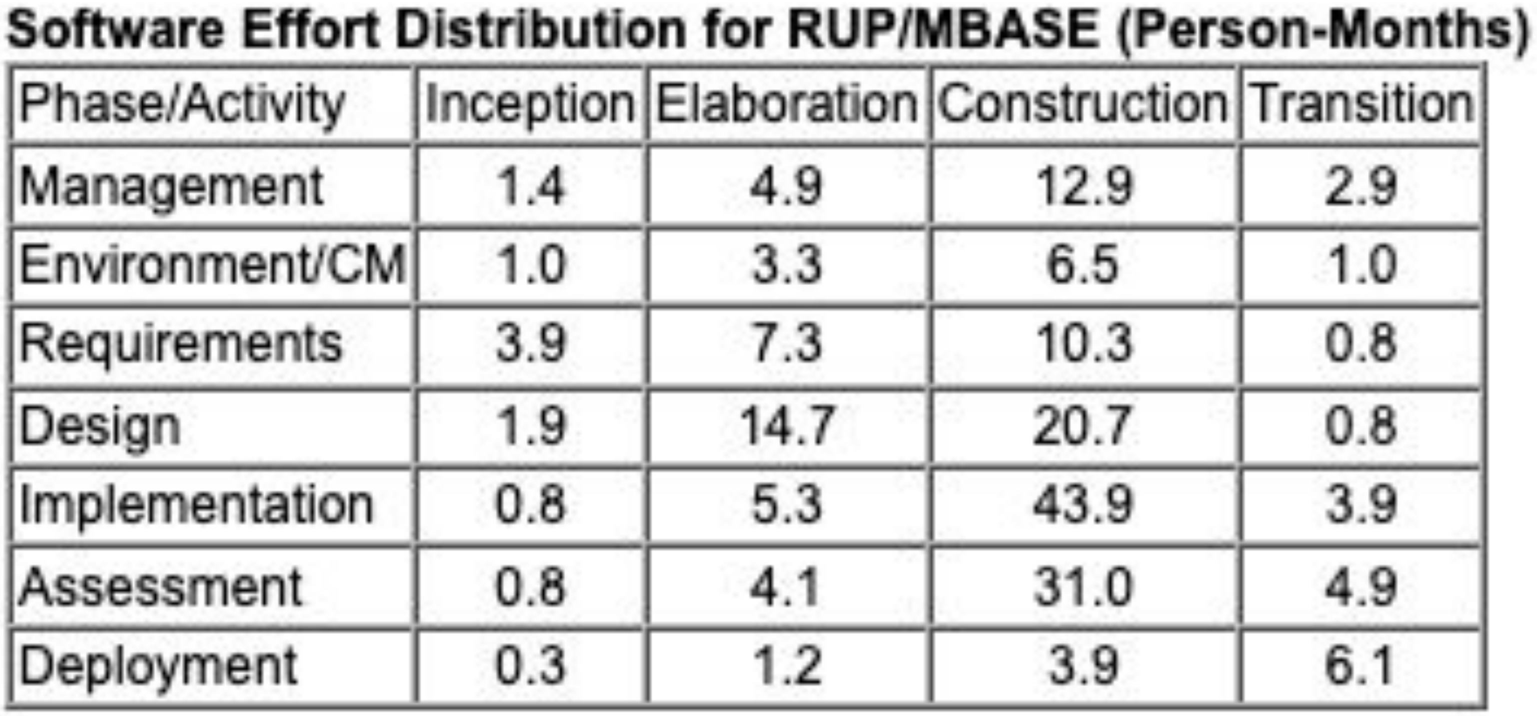
**Software Effort Distribution for RUP/MBASE (Person-Months)**

| Phase/Activity | Inception | Elaboration | Construction | Transition |
|---|---|---|---|---|
| Management | 1.4 | 4.9 | 12.9 | 2.9 |
| Environment/CM | 1.0 | 3.3 | 6.5 | 1.0 |
| Requirements | 3.9 | 7.3 | 10.3 | 0.8 |
| Design | 1.9 | 14.7 | 20.7 | 0.8 |
| Implementation | 0.8 | 5.3 | 43.9 | 3.9 |
| Assessment | 0.8 | 4.1 | 31.0 | 4.9 |
| Deployment | 0.3 | 1.2 | 3.9 | 6.1 |

**Staffing Profile**

https://csse.usc.edu/tools/COCOMOII.php

# COCOMO II Calculator (read chap 5 of the text book for details)



Software Effort Distribution for RUP/MBASE (Person-Months)

| Phase/Activity | Inception | Elaboration | Construction | Transition |
|---|---|---|---|---|
| Management | 1.4 | 4.9 | 12.9 | 2.9 |
| Environment/CM | 1.0 | 3.3 | 6.5 | 1.0 |
| Requirements | 3.9 | 7.3 | 10.3 | 0.8 |
| Design | 1.9 | 14.7 | 20.7 | 0.8 |
| Implementation | 0.8 | 5.3 | 43.9 | 3.9 |
| Assessment | 0.8 | 4.1 | 31.0 | 4.9 |
| Deployment | 0.3 | 1.2 | 3.9 | 6.1 |

Note: Assume $ = INR for interpretation of the inputs and results;; i.e., 25k rupees average salary per month

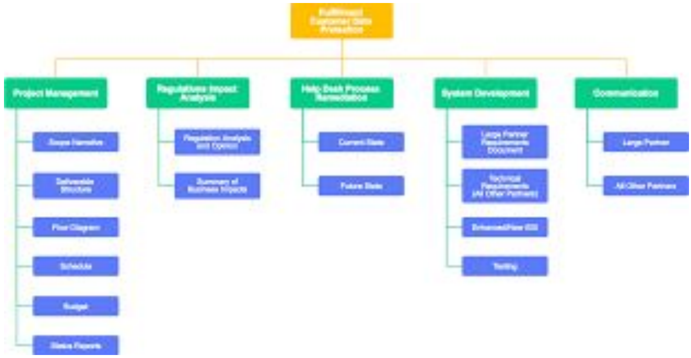
# CS6-#2 COCOMO II Calculator

Keeping the function points as 500, observe the effect of software scale and cost drivers on the estimated effort, schedule and cost.

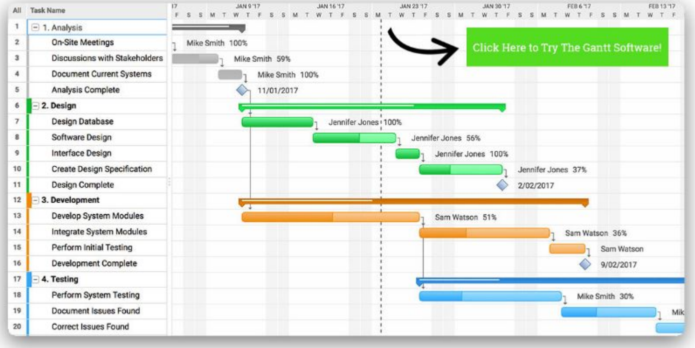
http://softwarecost.org/tools/COCOMO/

# Project Planning involves

- **Work Breakdown Structure** (WBS): Breaking down the work into parts and assign these to project team members, anticipate problems that might arise and prepare tentative solutions to those problems.


Fulfillment Customer Data Protection
Project Management
Scope Narrative
Deliverable Structure
Flow Diagram
Schedule
Budget
Status Reports
Regulations Impact Analysis
Regulation Analysis and Opinion
Summary of Business Impacts
Help Desk Process Remediation
Current State
Future State
System Development
Large Partner Requirements Document
Technical Requirements (All Other Partners)
Enhanced/New IDS
Testing
Communication
Large Partner
All Other Partners

- **Project Plan**, which is created at the start of a project, is used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.


17
JAN 9 '17
JAN 16 '17
JAN 23 '17
JAN 30 '17
FEB 6 '17
FEB 13 '17
F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F
Click Here to Try The Gantt Software!

| All | Task Name | |
|---|---|---|
| 1 | ⊟ 1. Analysis | |
| 2 | On-Site Meetings | Mike Smith 100% |
| 3 | Discussions with Stakeholders | Mike Smith 59% |
| 4 | Document Current Systems | Mike Smith 100% |
| 5 | Analysis Complete | 11/01/2017 |
| 6 | ⊟ 2. Design | |
| 7 | Design Database | Jennifer Jones 100% |
| 8 | Software Design | Jennifer Jones 56% |
| 9 | Interface Design | Jennifer Jones 100% |
| 10 | Create Design Specification | Jennifer Jones 37% |
| 11 | Design Complete | 2/02/2017 |
| 12 | ⊟ 3. Development | |
| 13 | Develop System Modules | Sam Watson 51% |
| 14 | Integrate System Modules | Sam Watson 36% |
| 15 | Perform Initial Testing | Sam Watson |
| 16 | Development Complete | 9/02/2017 |
| 17 | ⊟ 4. Testing | |
| 18 | Perform System Testing | Mike Smith 30% |
| 19 | Document Issues Found | Mik |
| 20 | Correct Issues Found | |

# Example of WBS fragment

- 3 Physical Design
  - 3.01 Design or specify physical database
    - 3.01.01 Review logical database design
    - 3.01.02 Determine access methods to be used
    - 3.01.03 Normalize database
    - 3.01.04 Design database architecture
    - 3.01.05 Identify reusable database structures
    - 3.01.06 Develop detailed database layout
    - 3.01.07 Develop database file, record, and schema descriptions
    - 3.01.08 Develop module calling sequences
    - 3.01.09 Update data dictionary entries
    - 3.01.10 Validate physical database design

See WBS waterfall.pdf on the course website for a complete decomposition

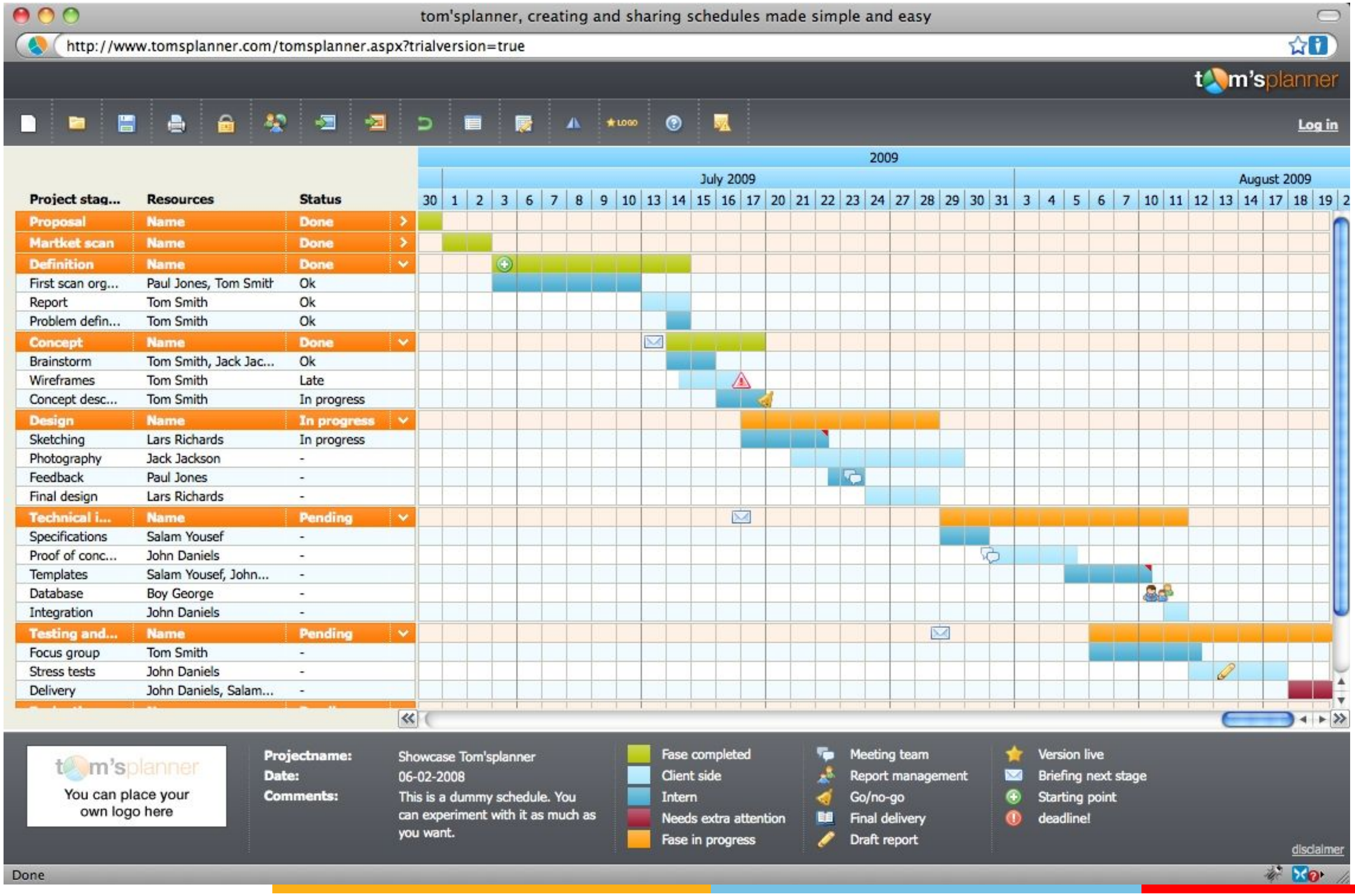
# CS6-#3 Gantt Chart

| Activity | Duration (weeks) | Depends on | ResouGrce |
|---|---|---|---|
| A | 3 | | SA |
| B | 1 | A | SD1 |
| C | 2 | A | SD2 |
| D | 4 | A | SD2 |
| E | 3 | B | CD1 |
| F | 3 | C | CD1 |
| G | 6 | D | CD2 |
| H | 3 | E,F,G | SA |

Prepare a Gantt Chart (on paper or any tool you can find) to find the critical path and project duration

# Activity planning – Sample Gantt Charting tool

## Tomsplanner (login using your BITS email ID)



tom'splanner, creating and sharing schedules made simple and easy

http://www.tomsplanner.com/tomsplanner.aspx?trialversion=true

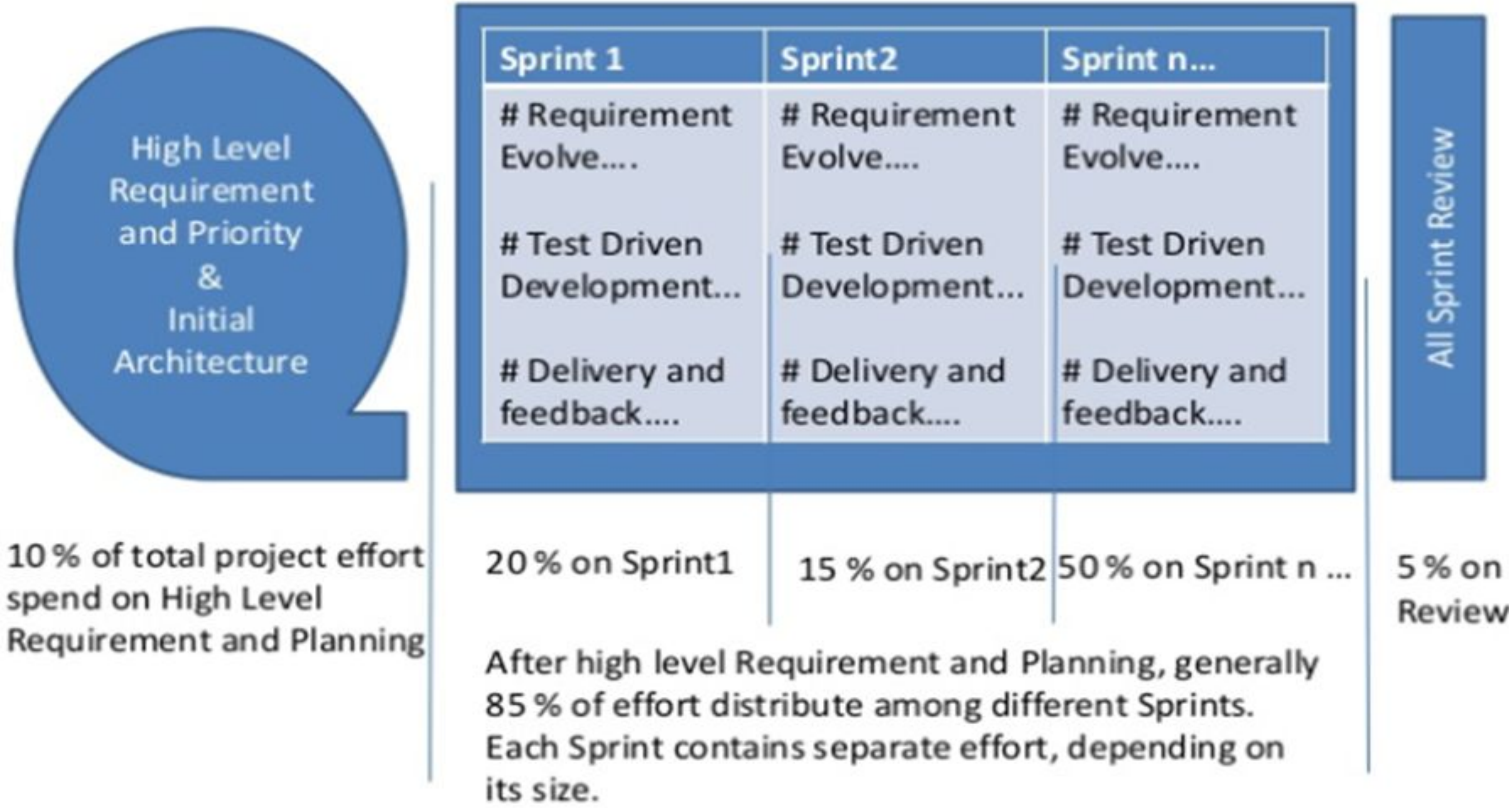
| Project stag... | Resources | Status |
|---|---|---|
| **Proposal** | **Name** | **Done** |
| **Martket scan** | **Name** | **Done** |
| **Definition** | **Name** | **Done** |
| First scan org... | Paul Jones, Tom Smitt | Ok |
| Report | Tom Smith | Ok |
| Problem defin... | Tom Smith | Ok |
| **Concept** | **Name** | **Done** |
| Brainstorm | Tom Smith, Jack Jac... | Ok |
| Wireframes | Tom Smith | Late |
| Concept desc... | Tom Smith | In progress |
| **Design** | **Name** | **In progress** |
| Sketching | Lars Richards | In progress |
| Photography | Jack Jackson | - |
| Feedback | Paul Jones | - |
| Final design | Lars Richards | - |
| **Technical i...** | **Name** | **Pending** |
| Specifications | Salam Yousef | - |
| Proof of conc... | John Daniels | - |
| Templates | Salam Yousef, John... | - |
| Database | Boy George | - |
| Integration | John Daniels | - |
| **Testing and...** | **Name** | **Pending** |
| Focus group | Tom Smith | - |
| Stress tests | John Daniels | - |
| Delivery | John Daniels, Salam... | - |

| | |
|---|---|
| **Projectname:** | Showcase Tom'splanner |
| **Date:** | 06-02-2008 |
| **Comments:** | This is a dummy schedule. You can experiment with it as much as you want. |

You can place your own logo here

| | | | | | |
|---|---|---|---|---|---|
| 🟩 | Fase completed | 💬 | Meeting team | ★ | Version live |
| ▢ | Client side | 👤 | Report management | ✉ | Briefing next stage |
| 🟦 | Intern | 📣 | Go/no-go | ⊕ | Starting point |
| 🟥 | Needs extra attention | 📖 | Final delivery | ! | deadline! |
| 🟧 | Fase in progress | ✎ | Draft report | | |

# Example of effort distribution in agile process (Scrum)



High Level Requirement and Priority & Initial Architecture

| Sprint 1 | Sprint2 | Sprint n... |
|---|---|---|
| # Requirement Evolve…. | # Requirement Evolve…. | # Requirement Evolve…. |
| # Test Driven Development... | # Test Driven Development... | # Test Driven Development... |
| # Delivery and feedback…. | # Delivery and feedback…. | # Delivery and feedback…. |

All Sprint Review

10 % of total project effort spend on High Level Requirement and Planning

20 % on Sprint1

15 % on Sprint2

50 % on Sprint n ...

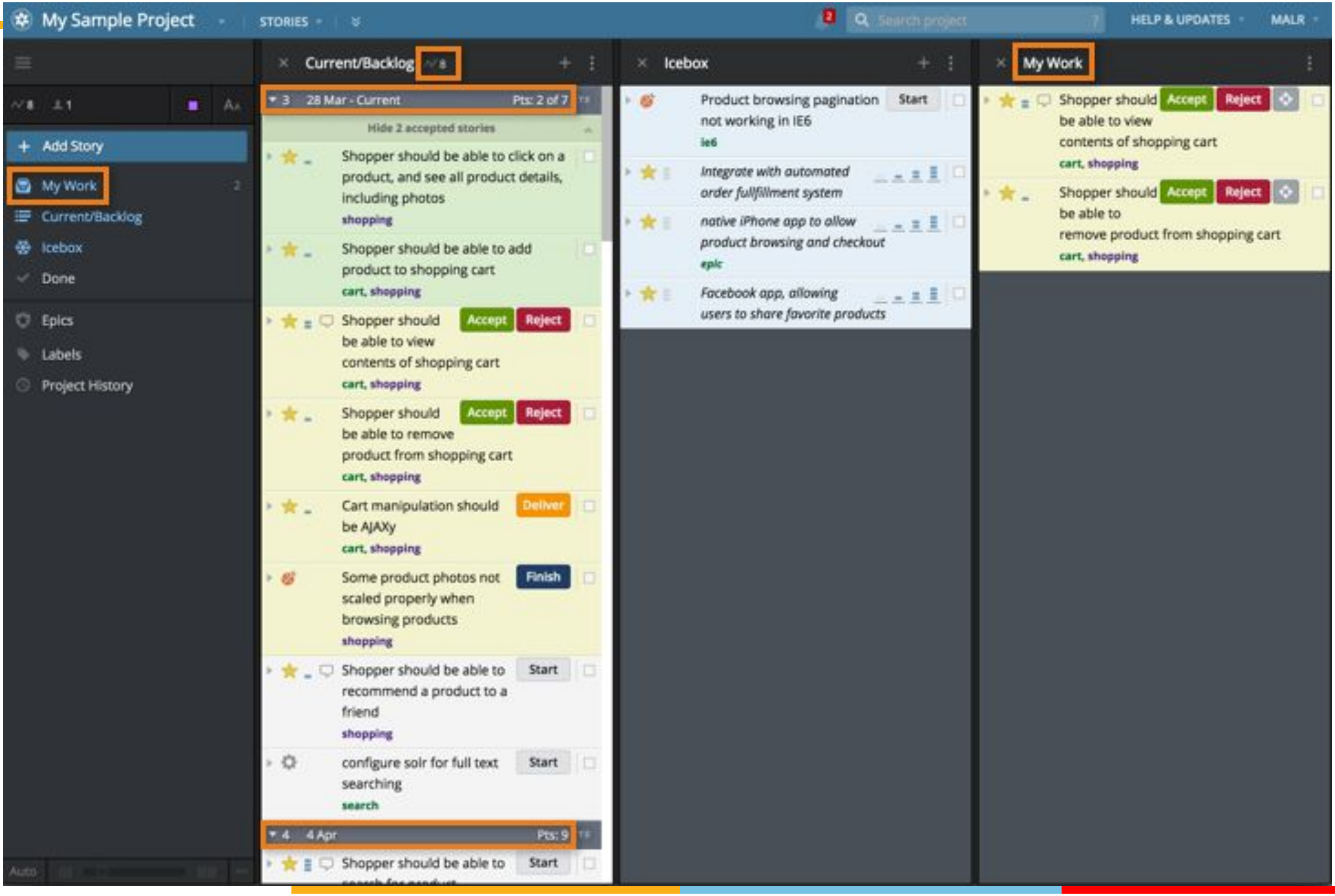
5 % on Review

After high level Requirement and Planning, generally 85 % of effort distribute among different Sprints. Each Sprint contains separate effort, depending on its size.

https://www.slideshare.net/brickedestimation/effort-distributiononwaterfallandagile

# Pivotal Tracker

https://www.pivotaltracker.com/help/articles/creating_a_project/



Current/Backlog

3 28 Mar - Current Pts: 2 of 7

Hide 2 accepted stories

Shopper should be able to click on a product, and see all product details, including photos
shopping

Shopper should be able to add product to shopping cart
cart, shopping

Shopper should be able to view contents of shopping cart [Accept] [Reject]
cart, shopping

Shopper should be able to remove product from shopping cart [Accept] [Reject]
cart, shopping

Cart manipulation should be AJAXy [Deliver]
cart, shopping

Some product photos not scaled properly when browsing products [Finish]
shopping

Shopper should be able to recommend a product to a friend [Start]
shopping

configure solr for full text searching [Start]
search

4 4 Apr Pts: 9

Shopper should be able to search for product [Start]

Icebox

Product browsing pagination not working in IE6 [Start]
ie6

*Integrate with automated order fulfillment system*

*native iPhone app to allow product browsing and checkout*
epic

*Facebook app, allowing users to share favorite products*

My Work

Shopper should be able to view contents of shopping cart [Accept] [Reject]
cart, shopping

Shopper should be able to remove product from shopping cart [Accept] [Reject]
cart, shopping

# Pivotal Tracker

- Epics
- Icebox
- My work
- Story types
  - Features
  - Chores
  - Bugs
  - Releases

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

# SS ZG622:

# Software Project Management

## Contact Session # 7

Prof. Narasimha Bolloju, BITS-Pilani, Hyderabad Campus

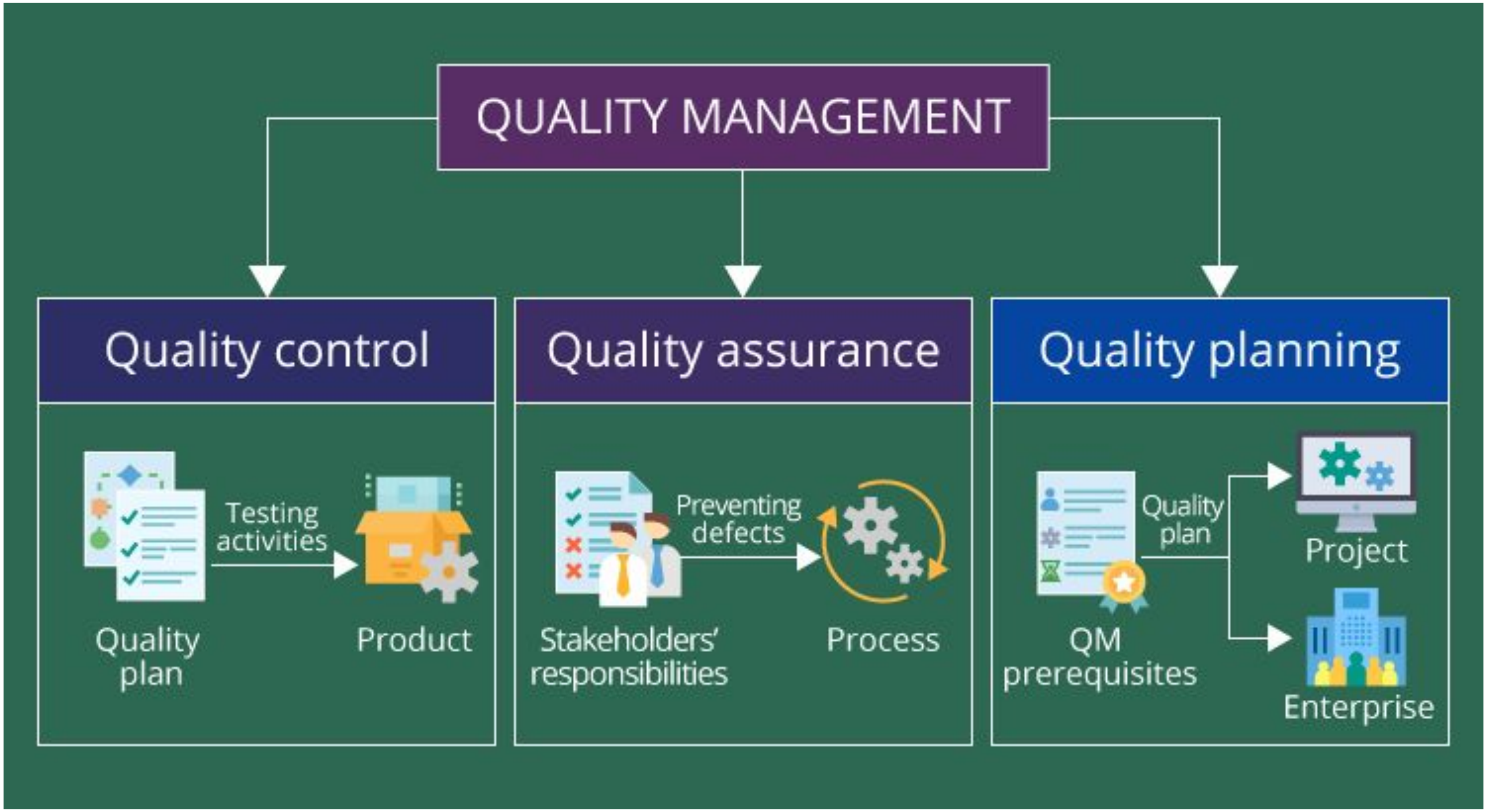
# Topics

- Quality management
- Quality planning
  - Identification relevant quality attributes
  - Identification of associated measures
  - Setting up standards for the product and processes

# Quality

- Quality, simplistically, means that a product should meet its specification
- The software product should deliver the required functionality (*functional requirements*) with the required **quality attributes** (*non–functional requirements*)

# Quality management



https://www.scnsoft.com/blog/quality-management-optimization

# Quality management

- Quality planning
  - setting quality standards and developing a plan to achieve them
- Quality assurance
  - ensuring that the development process is compliant with the quality plan
- Quality control
  - monitoring and evaluating the product to ensure that it meets the quality requirements

*Further discussion on this topic is planned during the second half of this course*

# Quality plan

- Identifies the most significant quality attributes appropriate for the product
- Defines the assessment process in detail for each quality attribute (including goals or target values)
- Indicates which standards should be applied and defines new standards as necessary

# A part of an example quality plan

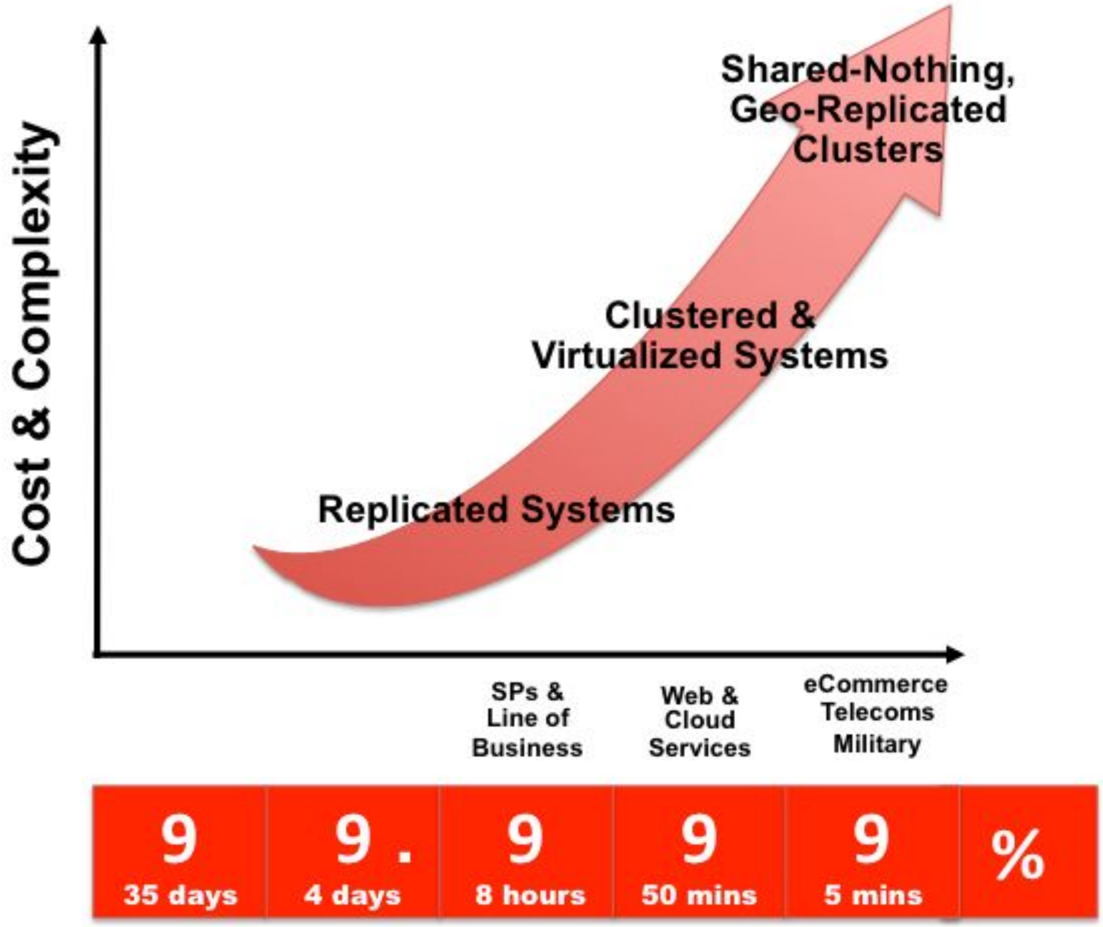
| Process | Process quality standards & Stakeholder expectations | Quality assurance activity | Frequency/ interval | Who is responsible |
|---|---|---|---|---|
| Review software development practices of software application XYZ | Developers have completely and accurately captured application requirements. | Peer review of software requirements specification. | At regular intervals during the collection of requirements and a final review at the conclusion of requirements collection. | Lead developer in conjunction with other knowledgeable developers. |
| | | | | |

Source: www.acqnotes.com

# Quality attribute example

- Reliability – probability of system working satisfactorily within a specific period of time
- Possible measures:
  - Availability: % of a time that a system is usable
  - Mean time between failures: total service time/number of failures
  - Failure on demand: probability that system will not available when needed
  - Support activity: number of fault reports generated and processed

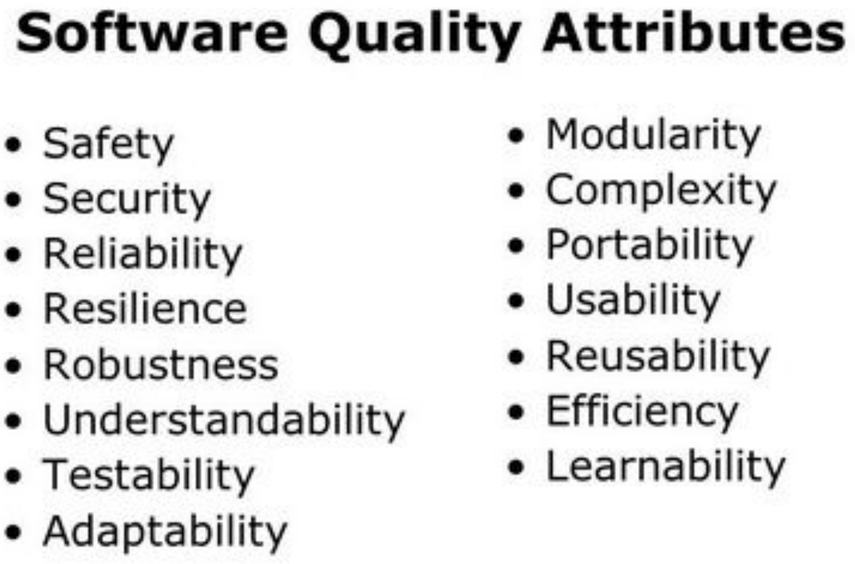
# Trade-offs: Cost & Complexity vs. Availability



https://docs.oracle.com/cd/E17952_01/mysql-5.5-en/ha-overview.html

# CS#7-1

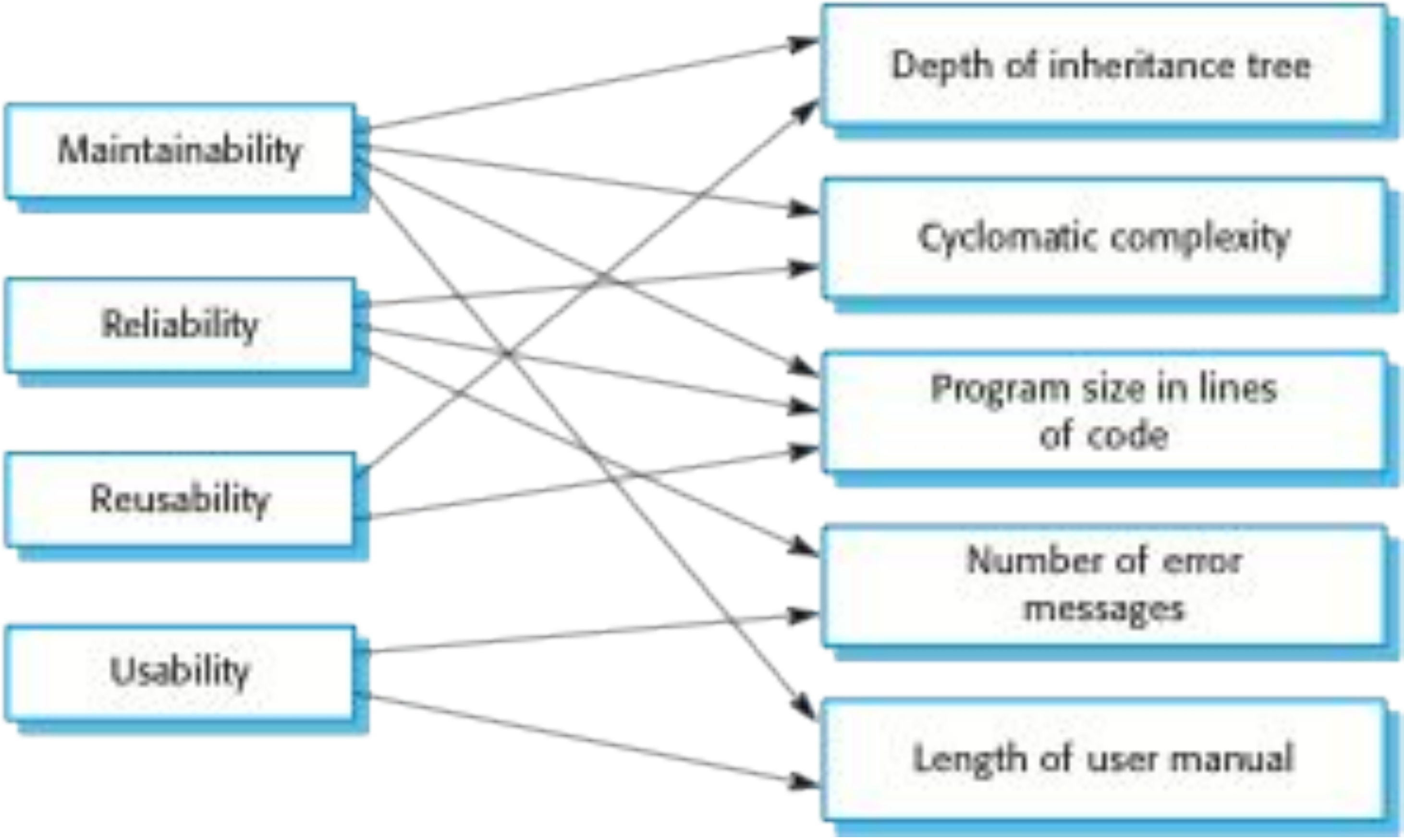
## Suggest and justify three quality attributes relevant for Mux-Core system



**Software Quality Attributes**

- Safety
- Security
- Reliability
- Resilience
- Robustness
- Understandability
- Testability
- Adaptability
- Modularity
- Complexity
- Portability
- Usability
- Reusability
- Efficiency
- Learnability

# Quality attributes and some related measures/metrics



# CS#7-2

Suggest measures for the selected quality attributes relevant for Mux-Core system

Ex: Reliability – availability, MTTF, MTTR, prob. failure on demand

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

# SS ZG622:

# Software Project Management

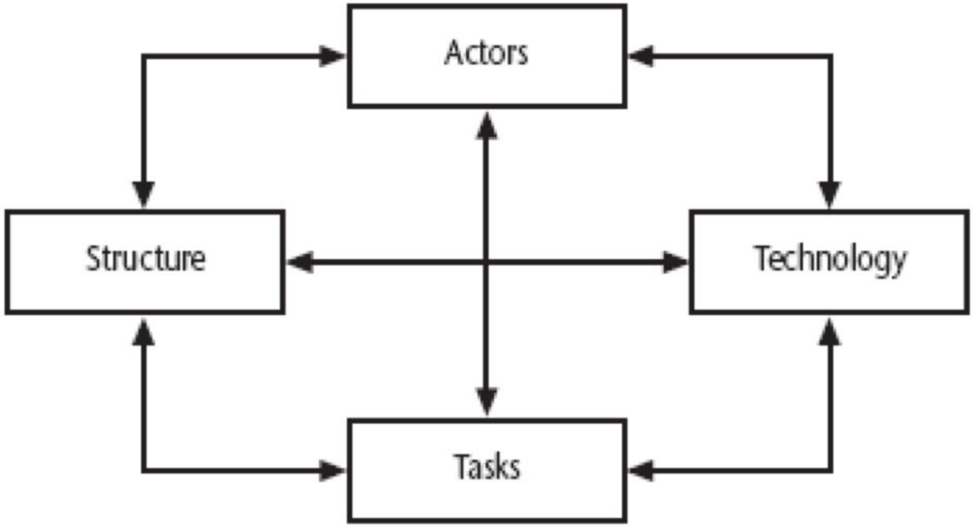
# Contact Session # 8

Prof. Narasimha Bolloju, BITS-Pilani, Hyderabad Campus

# Risk Management

# Risk Management

- Risk identification – what are the risks to a project?
- Risk analysis – which ones are really serious?
- Risk planning – what shall we do?
- Risk monitoring – has the planning worked?



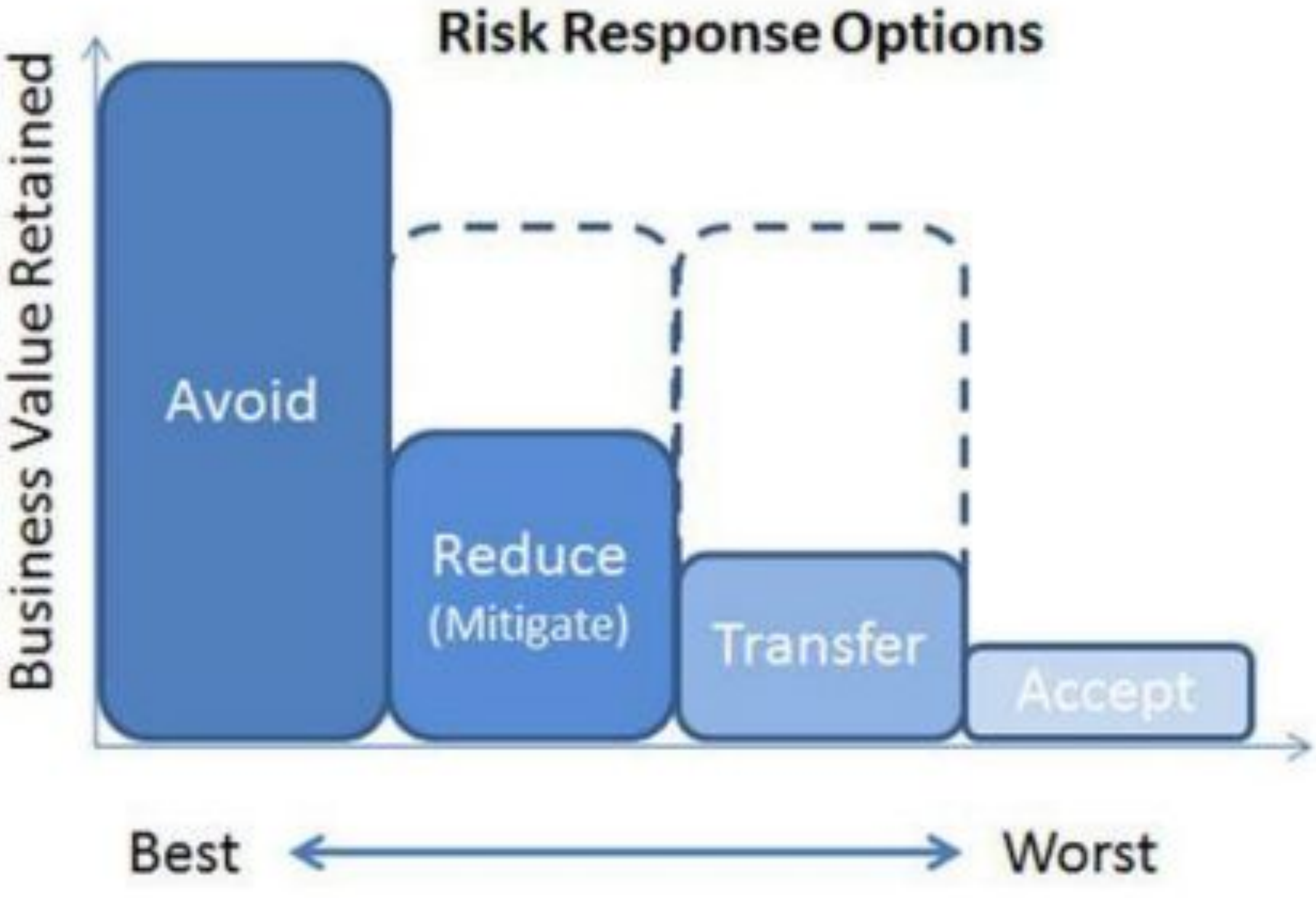
# Activity CS8#1: Identify possible risks for the Mux-Core project

Visit http://www.projectengineer.net/project-risk-checklist/ and identify five or six possible risks relevant for the Mux-Core project

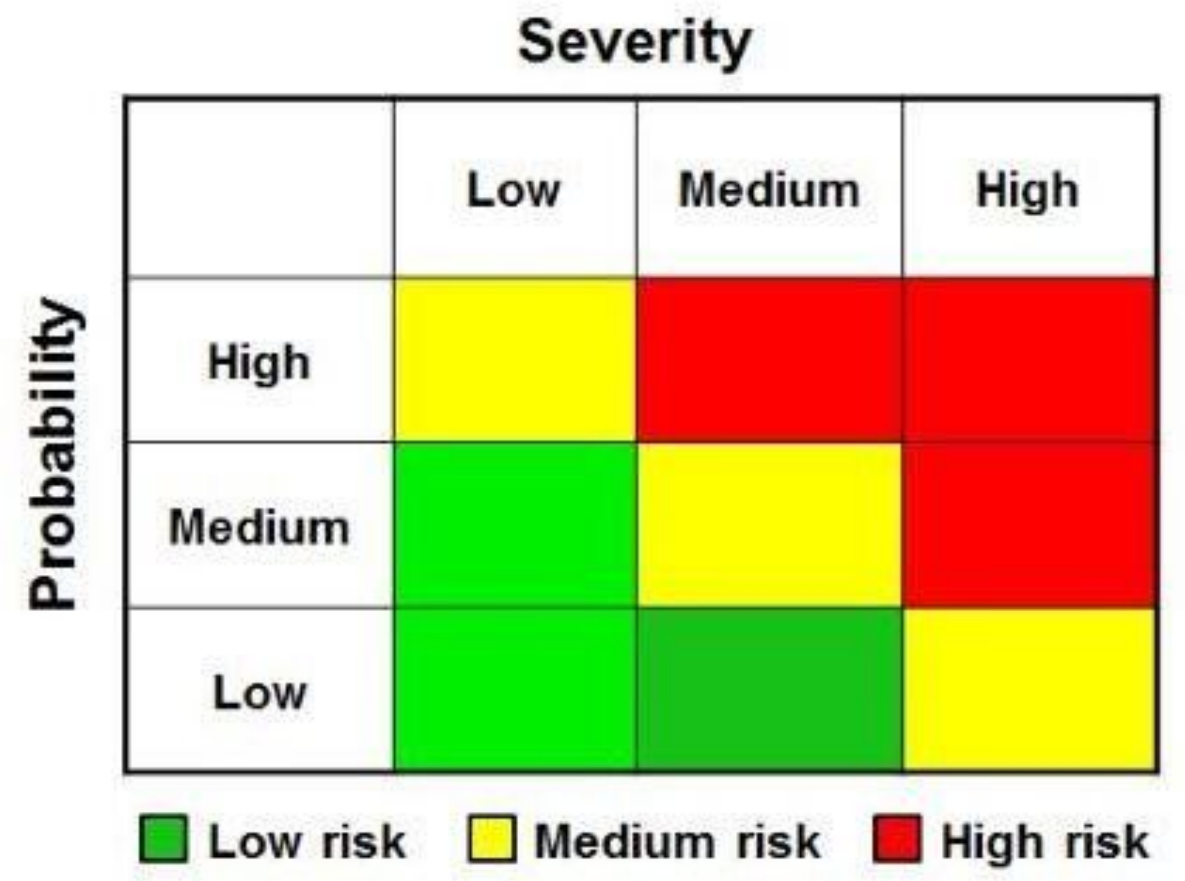
Post your suggestions with justification with the probability of its occurrence high/medium/low and severity or impact
(e.g., high > 0.8, medium > 0.4, low <= 0.4)

E.g., scope creep (medium) – many of the requirements related to administration and counter ticket sales are fairly well understood; but, the mobile/web for customers and management support need more clarity
(*Try different risk categories; Do not repeat the entries posted by others; instead improve those entries)*

# Risk Management



# Risk Probability and Severity



Severity

Probability

| | Low | Medium | High |
|---|---|---|---|
| High | 🟨 | 🟥 | 🟥 |
| Medium | 🟩 | 🟨 | 🟥 |
| Low | 🟩 | 🟩 | 🟨 |

🟩 Low risk 🟨 Medium risk 🟥 High risk

# Activity CS8#2: Suggest how the following risks can be managed

R1  - scope creep
R2 – quality of product does not meet standards
R3 – project is running behind schedule

Suggest **reduce (mitigate)** and **transfer** types of risk management for the above risks (refer your suggestions using R1, R2 or R3)

# DEMO –
# Risk Management Tools

A video is available at

https://www.softexpert.com/produto/risk-control-management/

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

# SS ZG622:

# Software Project Management

# Contact Session # 9

Prof. Narasimha Bolloju, BITS-Pilani, Hyderabad Campus

# Mid Semester Review

# Introduction to SPM

- Characteristics of software projects and challenges in managing projects
- Software project failures
- Project life cycle vs SDLC – differences and common activities
- Portfolio vs program vs project

# Project goals and metrics

- Business, program and project objectives
- Project management objectives vs. objectives for software system
- Integrating metrics within the software process

# Software development models (process models)

- Plan-driven vs agile process models
- Requirements engineering process
- Managing requirements (scope creep)
- Choosing the right process model for a given software development project

# Estimation techniques

- Challenges in effort estimation
- Bottom-up vs top-down estimation
- Function-point estimation
- Estimation in plan-driven vs agile processes (story points and Planning poker)
- Effort distribution in plan-driven vs agile processes

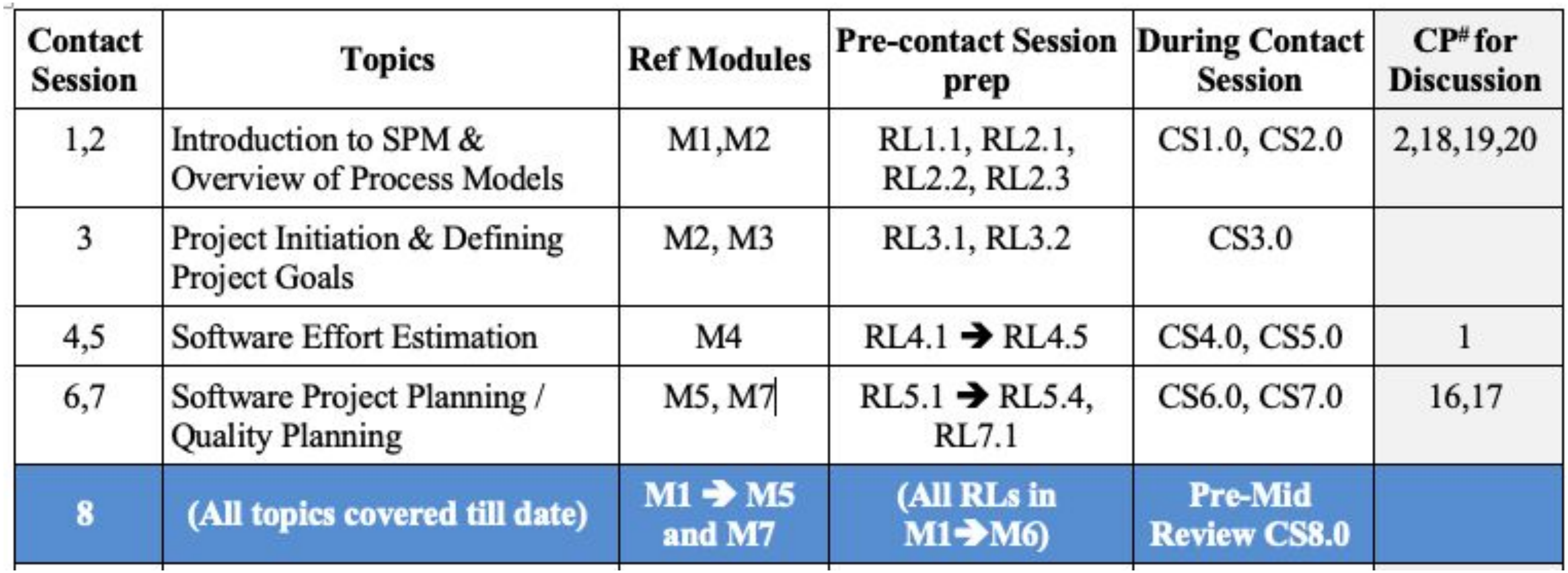
# Project planning and scheduling

- Phases in plan-driven processes
- Iterations in agile processes (sprints in Scrum)
- Work breakdown structure
- Scheduling using Gantt charts and AoN diagrams
- Resource scheduling
- Effort distribution in plan-driven and agile processes

# Quality planning

- Identification relevant (internal and external) quality attributes
- Selecting measures or metrics
- Defines the assessment process in detail for each quality attribute (including goals or target values)

# Syllabus for mid-semester test



| Contact Session | Topics | Ref Modules | Pre-contact Session prep | During Contact Session | CP$^{\#}$ for Discussion |
|---|---|---|---|---|---|
| 1,2 | Introduction to SPM & Overview of Process Models | M1,M2 | RL1.1, RL2.1, RL2.2, RL2.3 | CS1.0, CS2.0 | 2,18,19,20 |
| 3 | Project Initiation & Defining Project Goals | M2, M3 | RL3.1, RL3.2 | CS3.0 | |
| 4,5 | Software Effort Estimation | M4 | RL4.1 → RL4.5 | CS4.0, CS5.0 | 1 |
| 6,7 | Software Project Planning / Quality Planning | M5, M7 | RL5.1 → RL5.4, RL7.1 | CS6.0, CS7.0 | 16,17 |
| **8** | **(All topics covered till date)** | **M1 → M5 and M7** | **(All RLs in M1→M6)** | **Pre-Mid Review CS8.0** | |

Note: Risk management (RL6.1 and RL6.2) is not part of the mid-sem test syllabus