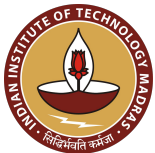


Software Engineering

Week 4: Software Project Management

Dr. Sridhar Iyer, IIT Bombay

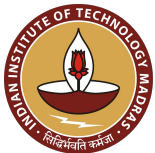
Dr. Prajish Prasad, FLAME University



Software Engineering

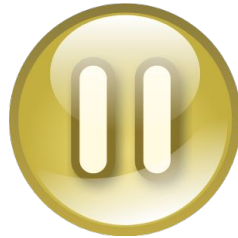
Project Management Overview

Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University

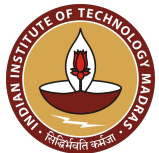


Reflection Spot

What do you think are the responsibilities of a project manager? What all is involved in managing a project?

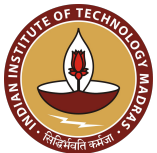


Please pause the video and written down your responses



Key Responsibilities of a Software Manager

- Main point of contact between clients and dev team
- Form and manage the dev team
- Project Scheduling
- Project Estimation
- Risk Management
- Configuration Management



Software Engineering

Project Estimation Techniques

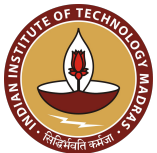
Dr. Sridhar Iyer, IIT Bombay

Dr. Prajish Prasad, FLAME University



Recap

- Practices involved in software project management
- This video - Project Estimation Techniques



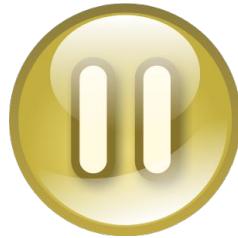
Importance of Estimation

- Establish cost
- Establish schedule
- Bidding for software projects - Cost and schedule must be provided to clients

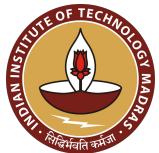


Reflection Spot

What can be estimated in a project? What are key parameter that you will provide an estimate for?



Please pause the video and written down your responses



Key Estimation Parameters

- Size of code
 - Number of lines of code
 - KLOC - number of 1000 lines of code
- Effort
 - How many people are required in the team
 - Person-month - effort an individual can typically put in a month
 - E.g. -
 - A project requires 12 person-month of development time
 - 4 developers - 3 months
 - 2 developers - 6 months



Project Estimation Techniques

- Empirical Estimation Techniques
 - Ask people who have completed similar projects
- Heuristic Techniques
 - Modelled using suitable mathematical expressions



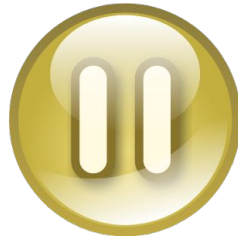
Empirical Estimation Techniques

- Expert Judgement
 - Educated guess
 - Estimate cost of different components
 - Combines the estimates



Reflection Spot

What are the drawbacks of using expert judgement for estimation?



Please pause the video and written down your responses

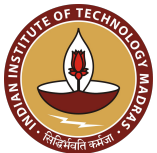
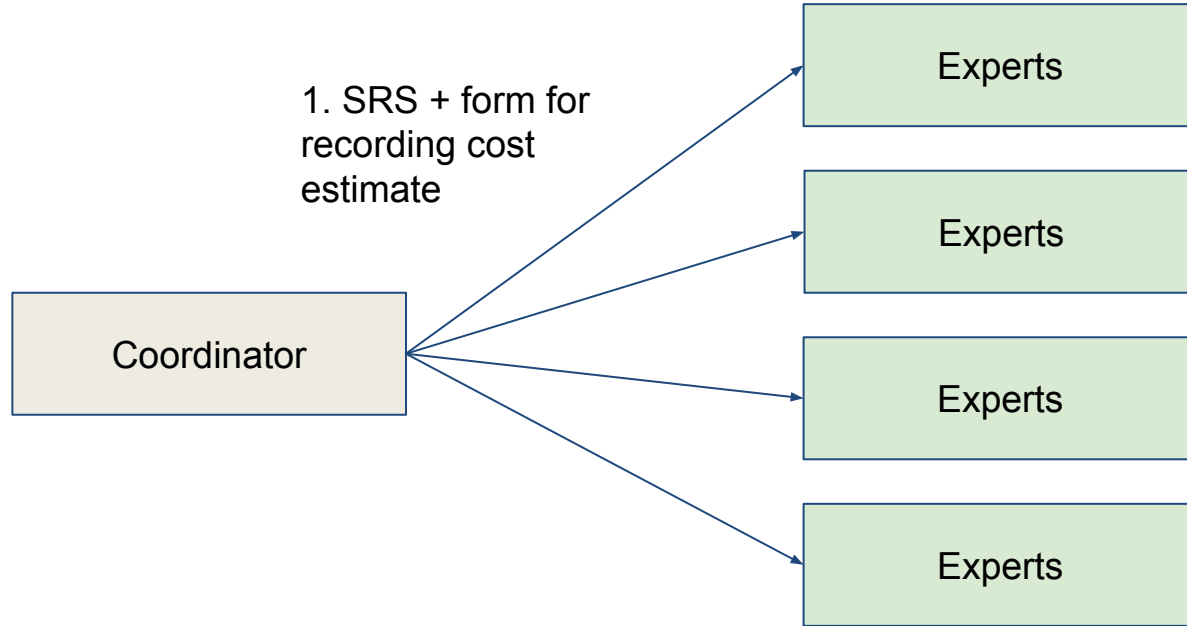


Drawbacks of Expert Judgement

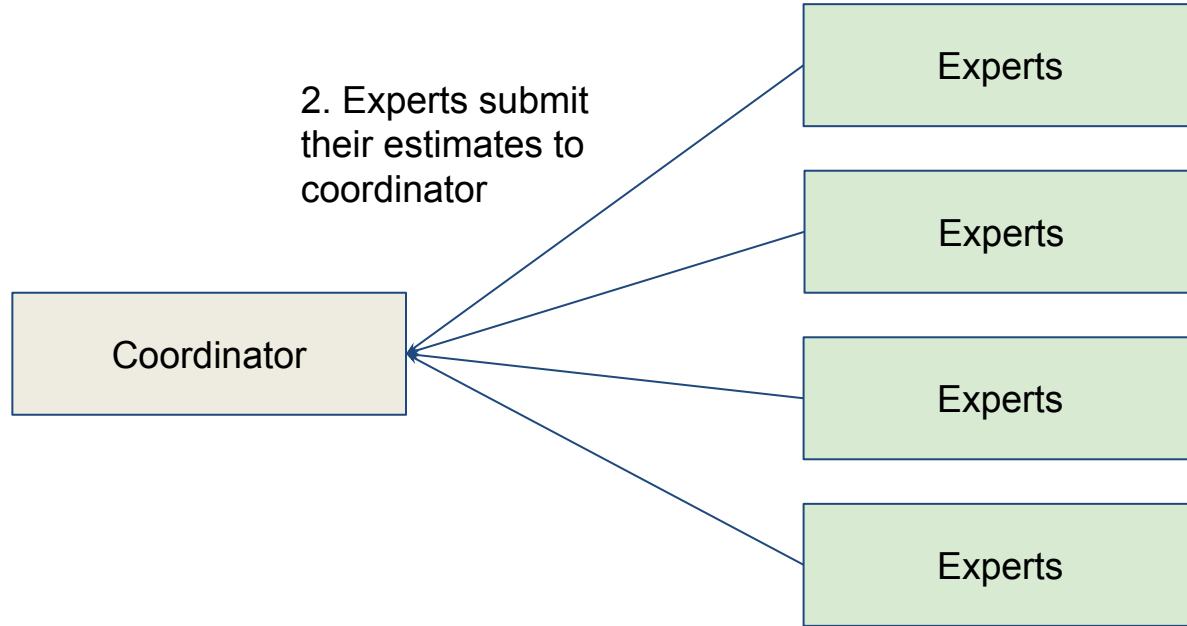
- Human errors, individual bias
- Optimistic estimates
- Overlook some factors, lack of adequate knowledge
- Solved by a group of experts?



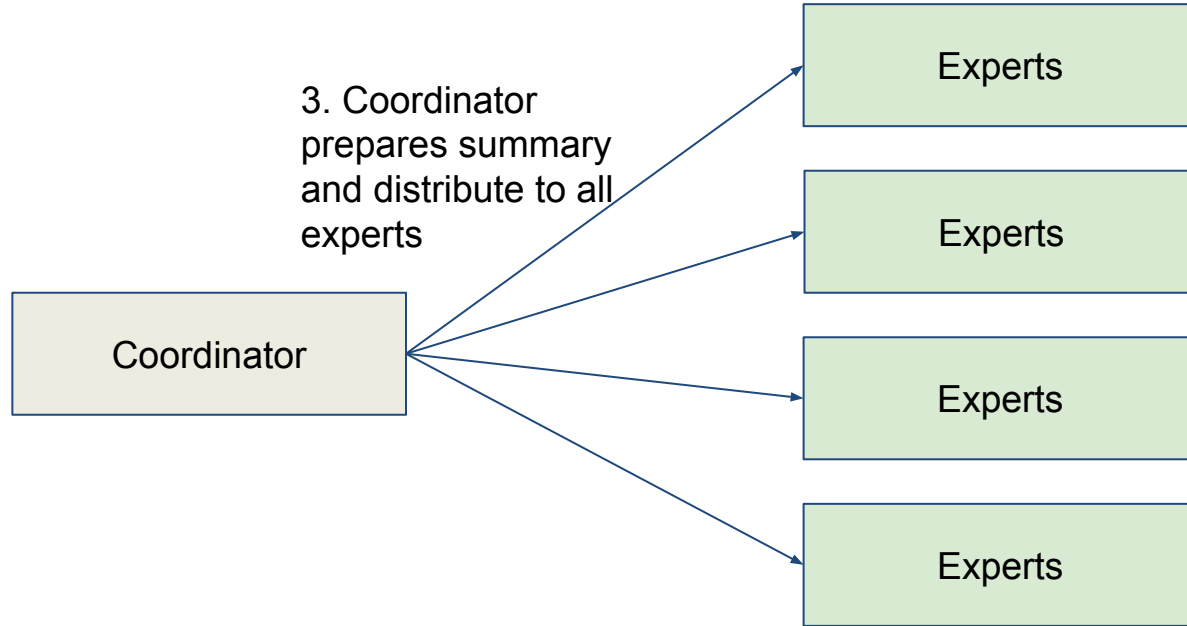
Delphi Technique



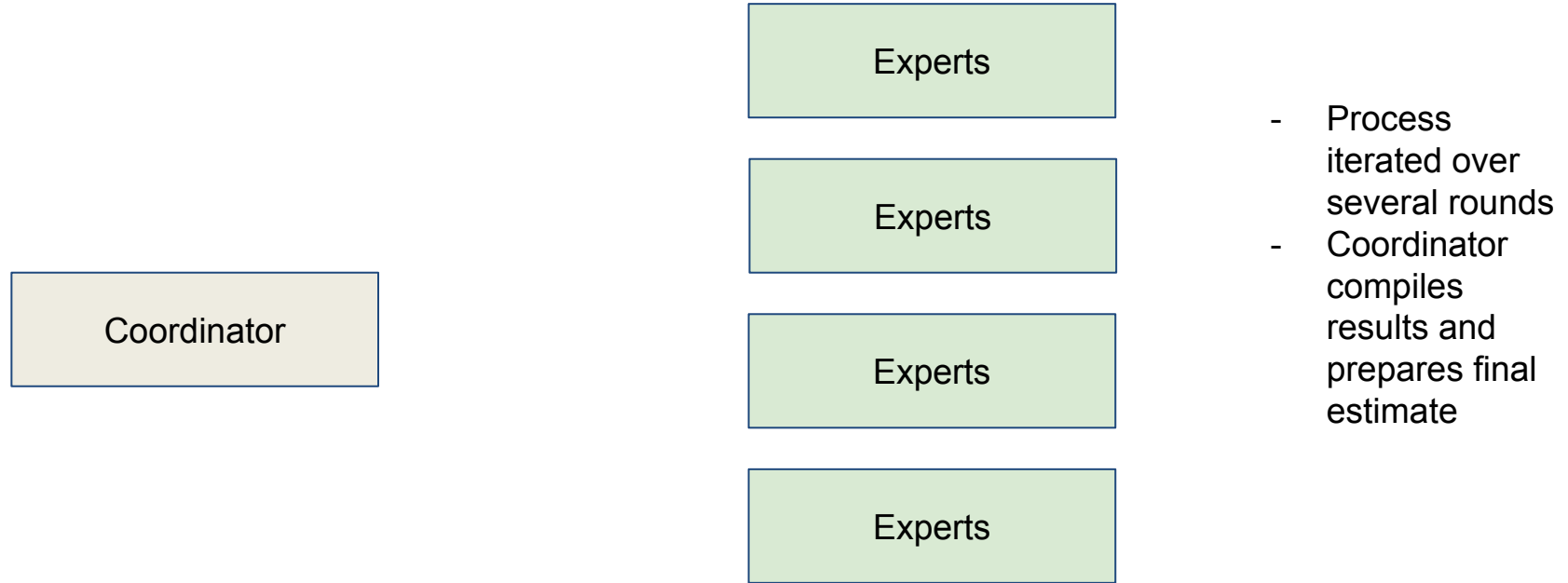
Delphi Technique



Delphi Technique



Delphi Technique



Project Estimation Techniques

- Empirical Estimation Techniques
 - Ask people who have completed similar projects
- **Heuristic Techniques**
 - Modelled using suitable mathematical expressions



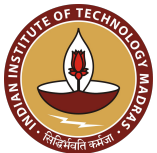
COCOMO Estimation Model

- Constructive Cost Estimation Model (COCOMO)
- Proposed by Boehm in 1981
- $\text{Effort} = a \times \text{SIZE}^b$



COCOMO Estimation Model

- $Effort = a \times SIZE^b$
- a and b depend on the type of project
 - Organic
 - Well-understood application program
 - Team size - small and experienced
 - Semi-detached
 - Mix of experienced + inexperienced people
 - Embedded
 - Strongly coupled with hardware



COCOMO Estimation Model

- Organic:

$$\text{Effort} = 2.4 \times (\text{SIZE in KLOC})^{1.05} \text{ Person Month}$$

- Semi-detached:

$$\text{Effort} = 3.0 \times (\text{SIZE in KLOC})^{1.12} \text{ Person Month}$$

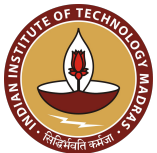
- Embedded:

$$\text{Effort} = 3.6 \times (\text{SIZE in KLOC})^{1.20} \text{ Person Month}$$



COCOMO Estimation Model: Example

- Amazon Seller Portal -
 - Type of system - semi-detached



COCOMO Estimation Model: Example

- Amazon Seller Portal -
 - Type of system - semi-detached
 - Estimate number of lines of code
 - 1. Add/Edit/Delete catalogue - 300 LOC
 - 2. Add/Edit/Delete inventory - 300 LOC
 - 3. Track orders - 1000 LOC
 - 4. Track payments - 800 LOC
 - 5. Track inventory - 800 LOC
 - 6. Track sales - specific day, last x days - 500 LOC
 - 7. Track customer feedback - 300 LOC
 - Total - 4000 LOC = 4 KLOC



COCOMO Estimation Model: Example

- Amazon Seller Portal -

- Type of system - semi-detached

Effort = $3.0 \times (\text{SIZE in KLOC})^{1.12}$ Person Month

- Estimate number of lines of code - 4 KLOC

○ **Effort = $3.0 \times (4)^{1.12} = 15.83$ Person Months - Initial Estimate**



COCOMO Estimation Model: Example

- Amazon Seller Portal -

- Type of system - semi-detached

Effort = $3.0 \times (\text{SIZE in KLOC})^{1.12}$ Person Month

- Estimate number of lines of code - 4 KLOC

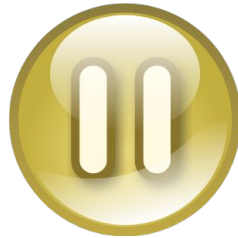
○ **Effort = $3.0 \times (4)^{1.12} = 15.83$ Person Months - Initial Estimate**

- **Cost = $16 \times 75,000 = \text{Rs. } 12 \text{ lakhs}$**

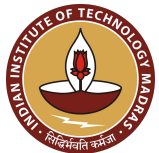


Reflection Spot

Apart from the size of the code, what other factors have to be considered to estimate **effort required for a project?**



Please pause the video and written down your responses



Effort Estimation Parameters

- People working in the project
- Technical attributes of the project
- Tools and practices used by the team



Cost driver attributes

Cost Drivers	Rating				
	Very Low	Low	Nom-inal	High	Very High
Product Attributes					
RELY, required reliability	.75	.88	1.00	1.15	1.40
DATA, database size		.94	1.00	1.08	1.16
CPLX, product complexity	.70	.85	1.00	1.15	1.30
Computer Attributes					
TIME, execution time constraint			1.00	1.11	1.30
STOR, main storage constraint			1.00	1.06	1.21
VITR, virtual machine volatility		.87	1.00	1.15	1.30
TURN, computer turnaround time		.87	1.00	1.07	1.15
Personnel Attributes					
ACAP, analyst capability	1.46	1.19	1.00	.86	.71
AEXP, application exp.	1.29	1.13	1.00	.91	.82
PCAP, programmer capability	1.42	1.17	1.00	.86	.70
VEXP, virtual machine exp.	1.21	1.10	1.00	.90	
LEXP, prog. language exp.	1.14	1.07	1.00	.95	
Project Attributes					
MODP, modern prog. practices	1.24	1.10	1.00	.91	.82
TOOL, use of SW tools	1.24	1.10	1.00	.91	.83
SCHED, development schedule	1.23	1.08	1.00	1.04	1.10

Taken from - A Concise Introduction to Software Engineering - Pankaj Jalote -
Table 4.1 pg 72



Cost driver attributes

Cost Drivers	Rating				
	Very Low	Low	Nom-inal	High	Very High
Product Attributes					
RELY, required reliability	.75	.88	1.00	1.15	1.40
DATA, database size		.94	1.00	1.08	1.16
CPLX, product complexity	.70	.85	1.00	1.15	1.30
Computer Attributes					
TIME, execution time constraint			1.00	1.11	1.30
STOR, main storage constraint			1.00	1.06	1.21
VITR, virtual machine volatility		.87	1.00	1.15	1.30
TURN, computer turnaround time		.87	1.00	1.07	1.15
Personnel Attributes					
ACAP, analyst capability	1.46	1.19	1.00	.86	.71
AEXP, application exp.	1.29	1.13	1.00	.91	.82
PCAP, programmer capability	1.42	1.17	1.00	.86	.70
VEXP, virtual machine exp.	1.21	1.10	1.00	.90	
LEXP, prog. language exp.	1.14	1.07	1.00	.95	
Project Attributes					
MODP, modern prog. practices	1.24	1.10	1.00	.91	.82
TOOL, use of SW tools	1.24	1.10	1.00	.91	.83
SCHED, development schedule	1.23	1.08	1.00	1.04	1.10

Taken from - A Concise Introduction to Software Engineering - Pankaj Jalote -
Table 4.1 pg 72



Cost driver attributes

Cost Drivers	Rating				
	Very Low	Low	Nom-inal	High	Very High
Product Attributes					
RELY, required reliability	.75	.88	1.00	1.15	1.40
DATA, database size		.94	1.00	1.08	1.16
CPLX, product complexity	.70	.85	1.00	1.15	1.30
Computer Attributes					
TIME, execution time constraint			1.00	1.11	1.30
STOR, main storage constraint			1.00	1.06	1.21
VITR, virtual machine volatility		.87	1.00	1.15	1.30
TURN, computer turnaround time		.87	1.00	1.07	1.15
Personnel Attributes					
ACAP, analyst capability	1.46	1.19	1.00	.86	.71
AEXP, application exp.	1.29	1.13	1.00	.91	.82
PCAP, programmer capability	1.42	1.17	1.00	.86	.70
VEXP, virtual machine exp.	1.21	1.10	1.00	.90	
LEXP, prog. language exp.	1.14	1.07	1.00	.95	
Project Attributes					
MODP, modern prog. practices	1.24	1.10	1.00	.91	.82
TOOL, use of SW tools	1.24	1.10	1.00	.91	.83
SCHED, development schedule	1.23	1.08	1.00	1.04	1.10

- Initial estimate = 15.83 PM
- Effort adjustment factor = $1.40 \times 1.08 \times 1.13 \times 0.95 = 1.62$
- Final estimate = $15.83 \times 1.62 = 25.65$ PM



Taken from - A Concise Introduction to Software Engineering - Pankaj Jalote - Table 4.1 pg 72

COCOMO Model - Summary

- Determine the type of product to be built
- Estimate the LOC
- Initial estimate
- Effort adjustment factor - 15 cost drivers
- Overall estimate



Summary

- Estimation Techniques
 - Empirical
 - Heuristic
- Which technique to use?
 - Depends on type of organization, type of product
- 92% of project managers made their estimates using experience instead of formulas^[1]



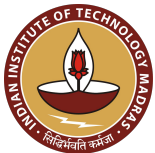
[1] A. Taylor. IT projects sink or swim. BCS Review, Jan. 2000.

Software Engineering

Project Scheduling

Dr. Sridhar Iyer, IIT Bombay

Dr. Prajish Prasad, FLAME University



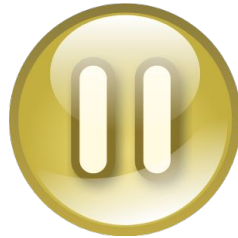
Project Scheduling

- Importance of a schedule
 - Monitor timely completion of task
 - Take corrective action

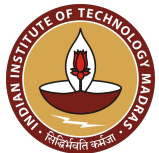


Reflection Spot

How do you go about creating a schedule for a project? What are the main activities involved in creating a schedule?

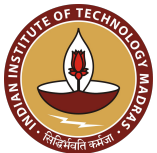


Please pause the video and written down your responses



Main Activities in Scheduling

- Identify all major activities
- Break down each activity into tasks
- Determine the dependency among different tasks
- Estimations for time durations required to complete the tasks
- Represent this information - chart, graph, network etc.
- Determine task starting and end dates from the representation
- Determine the critical path - a chain of tasks that determine the duration of the project.
- Allocate resources to tasks

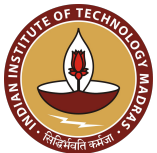
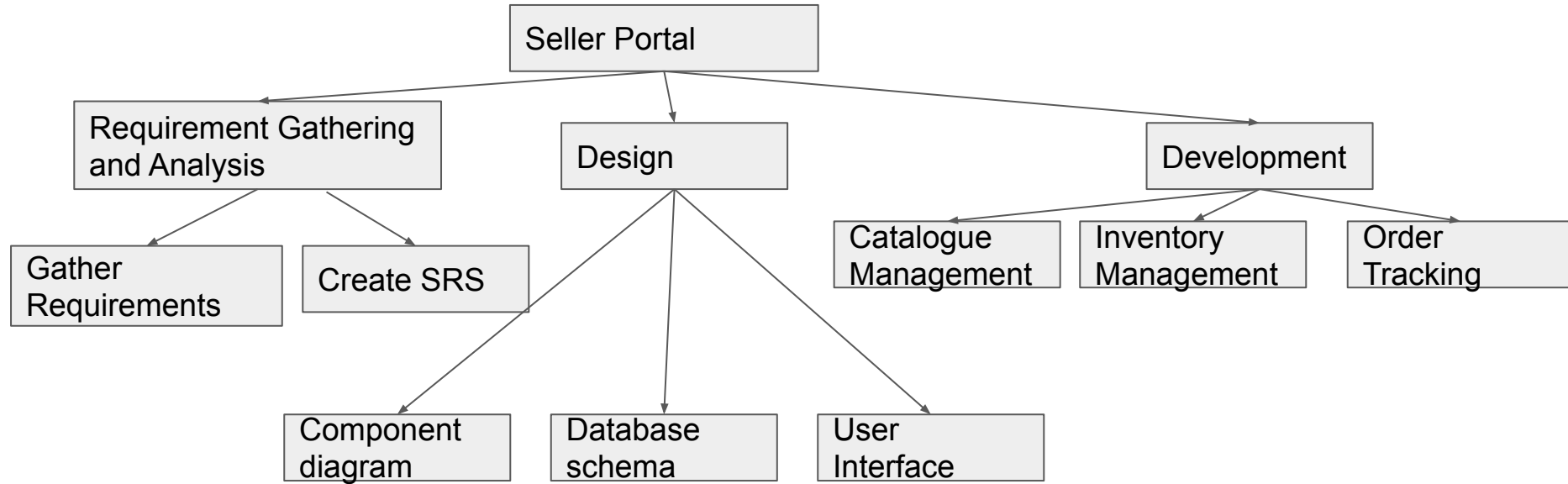


Breakdown activities into tasks

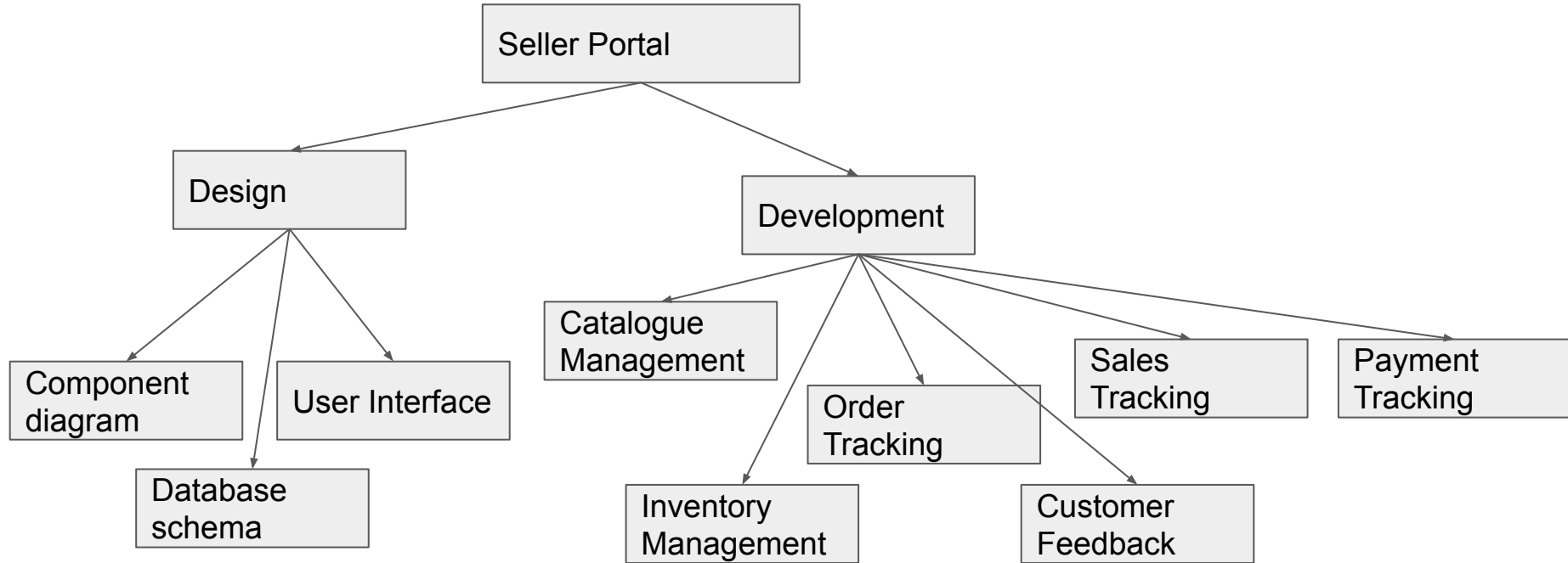
- Work breakdown structure (WBS)
- Create a tree like structure -
 - Root - Project name
 - Each node is broken down into smaller activities - children
 - Each leaf represents a task which can be allocated to a developer and scheduled
 - Task - roughly two weeks to develop



Example - Seller Portal



Example - Seller Portal

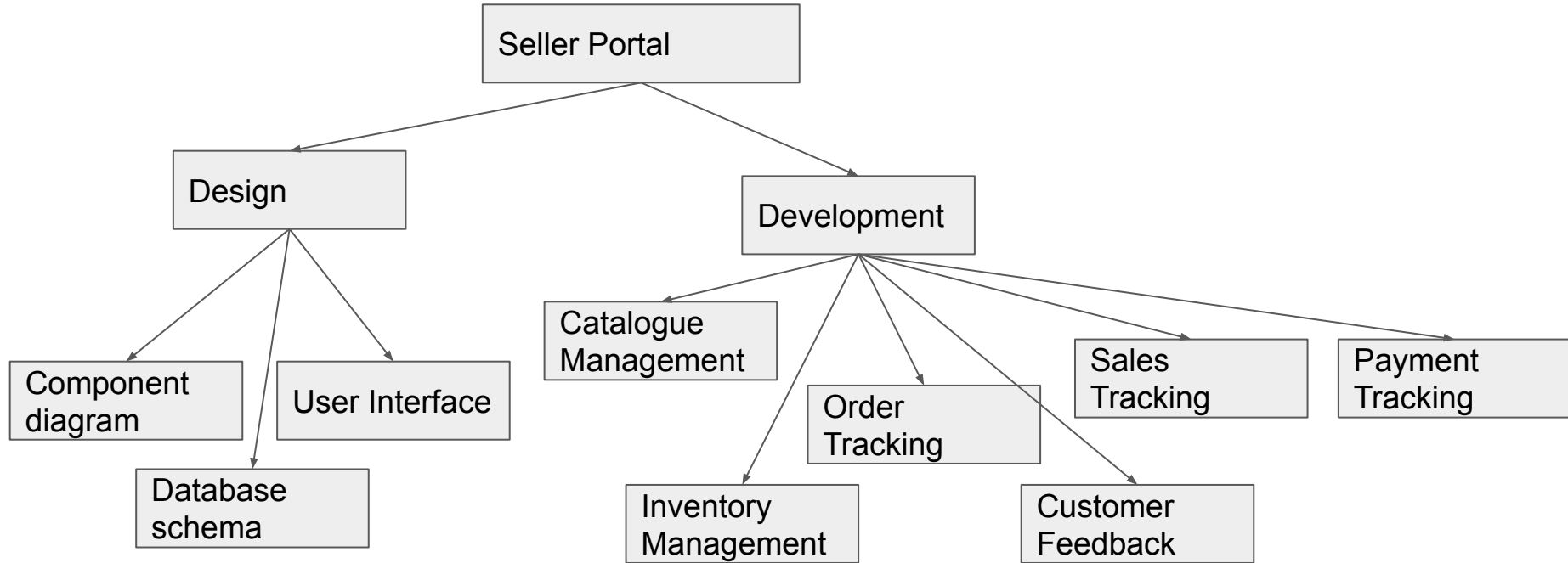


Activity Network

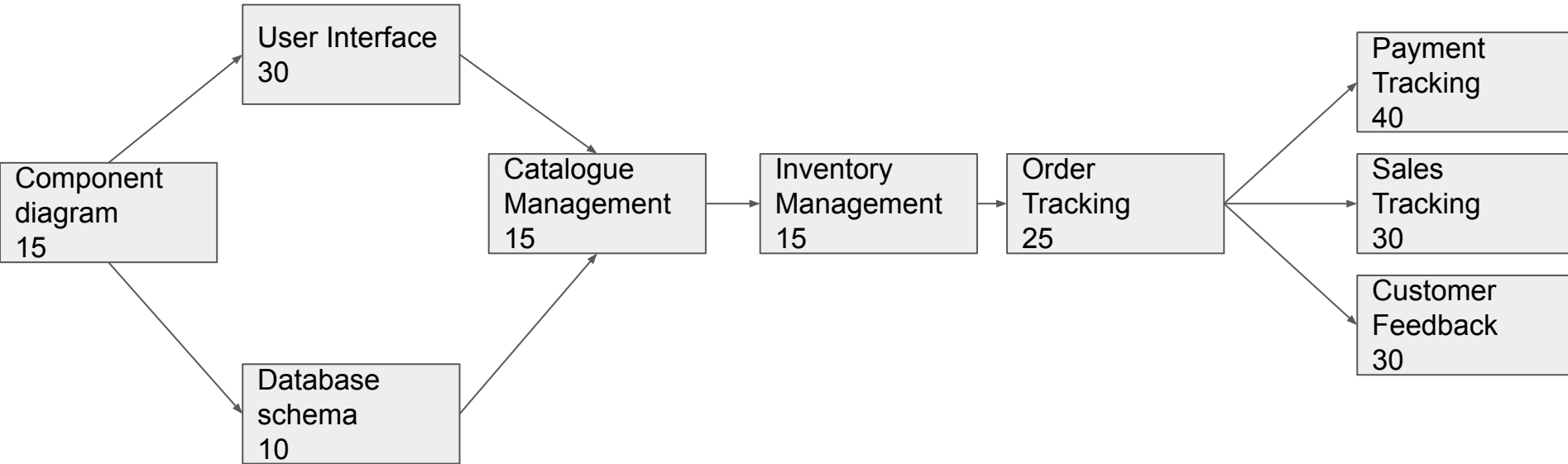
- Different activities making up a project, estimated durations, interdependencies
- Leaf nodes of the WBS become nodes of the activity network



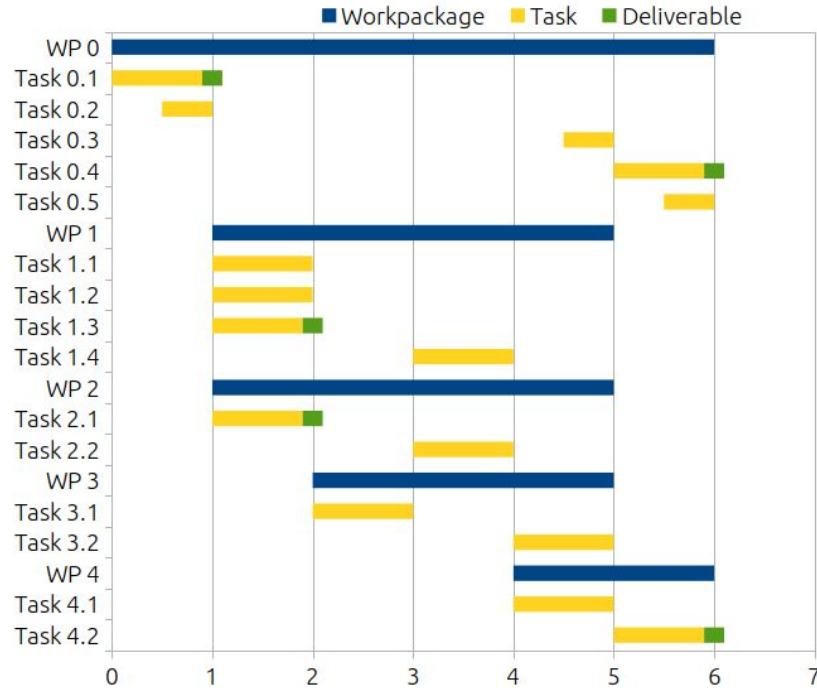
Example - Seller Portal



Activity Network Example



Gantt Chart



Main Activities in Scheduling

- Identify all major activities
- Break down each activity into tasks
- Determine the dependency among different tasks
- Estimations for time durations required to complete the tasks
- Represent this information - chart, graph, network etc.
- Determine task starting and end dates from the representation
- Determine the critical path - a chain of tasks that determine the duration of the project.
- Allocate resources to tasks



Software Engineering

Risk Management

Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University



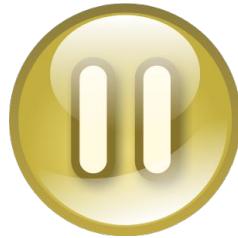
Risk Management

- A risk is an anticipated unfavourable event or circumstance that can occur while a project is underway
- Intangible nature of software
 - Syntax errors
 - Third party modules
- Conflicts in a team

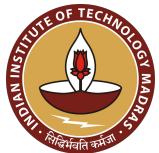


Reflection Spot

What are potential risks you can think of in the Amazon Seller portal system?



Please pause the video and written down your responses



Technical Risks

- Technical aspects of the project
- Due to development team's insufficient knowledge about the product



Technical Risk - Example

- Developing the wrong functions and user interfaces
- Mitigate risk
 - Communicate with clients, build prototypes



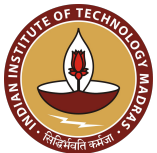
Technical Risk - Example

- Shortcomings in external components
- Mitigate risk
 - Benchmarking, regular inspections



Project Risks

Project risks occur due to problems in budget, schedule, personnel, resources, and customer-related problems



Project Risks - Example

- Schedule Slippage
- Mitigate this risk
 - Detailed milestones
 - Constant iterations
 - Communicate frequently with clients



Project Risks - Example

- Insufficient domain knowledge/technical knowledge
- Mitigate risk -
 - Hire developers with relevant experience, within the company, outside the company
 - Outsource to third party vendors



Project Risks - Example

- Personnel shortfall
- Mitigate risk -
 - Cross training - train multiple people with skills required to work on the project



Business Risk

- Risks which can harm the business aspects of the software product
- Example - Product not competitive in the market
- Mitigate risk -
 - Explore market for similar products



Business Risk Examples

- Gold plating - Developing unnecessary features
- Mitigate risk -
 - Communicate with clients
 - Cost-benefit analysis



Risk Identification and Mitigation

- Technical risks - risks related to the technical aspects of the project
- Project risks - occur due to problems in budget, schedule, personnel, resources, and customer-related problems
- Business risks - risks which can harm the business aspects of the software product



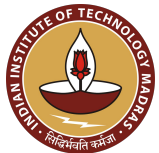
Risk Assessment

- Project manager creates a “risk table”
- Assigns probability (P)
- Impact (I) - negligible, marginal, critical, catastrophic (1-4)
- $Risk = P \times I$
- Sort the table in descending order



Risk Assessment - Example

Risk	Probability	Impact	
Schedule slippage	60%	3	1.8
Lack of experience building ecommerce applications	60%	3	1.8
Personnel shortfall	40%	3	1.2
Database not able to scale for large number of users	20%	2	0.4



Software Engineering

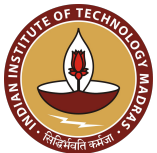
Project Management in Agile

Dr. Sridhar Iyer, IIT Bombay
Dr. Prajish Prasad, FLAME University



Recap

- Key activities in managing a software project - Plan and document perspective
- Agile perspective - does not predict cost and schedule at the start of the project



Agile Perspective

- Divided into iterations - 1-2 weeks
- User stories implemented in each iteration
- User stories are prioritized for next iteration



Team formation

- Size: 4-9 people
- Organize development in this team?
- Scrum -
 - Sprint - short, time-boxed period when a scrum team works to complete a set amount of work



Scrum Team

- Development Team - whoever is required to complete work in that given sprint
- Product Owner - interfaces between the client and the development team
- Scrum Master - ensures all activities are being done well



Sprint Planning

- Collaborative event - product owner, scrum master, development team
- Two basic questions
 - What work can get done in this sprint?
 - How will the chosen work get done?
- Sprint planning meeting - 2 hours per week of iteration



Product Backlog

- Prioritized list of work for the development team that is derived from user stories and requirements
- Who prioritises these items?
 - Sprint planning meeting



Example - Product Backlog

1. Catalogue Management
 - a. Add catalogue item
 - b. Edit catalogue item
 - c. Delete catalogue item
2. Inventory Management
 - a. Add item
 - b. Edit item
 - c. Delete item
3. Order tracking
4. Payment Tracking
5. Sales Tracking
6. Customer Feedback



▼ Backlog (10 issues)

0 0 0 Create sprint

IITMSE-1 Add Catalogue TO DO ▾

IITMSE-2 Edit Catalogue TO DO ▾

IITMSE-3 Delete Catalogue TO DO ▾

IITMSE-4 Add Inventory Item TO DO ▾

IITMSE-5 Edit Inventory Item TO DO ▾

IITMSE-6 Delete Inventory Item TO DO ▾

IITMSE-7 Order tracking TO DO ▾

IITMSE-8 Sales tracking TO DO ▾

IITMSE-9 Payment tracking TO DO ▾

IITMSE-10 Customer Feedback TO DO ▾

▼ **IITMSE Sprint 1** 2 Aug – 16 Aug (2 issues)

0 0 0

Complete sprint



 IITMSE-1 Add Catalogue

TO DO ▼



 IITMSE-4 Add Inventory Item

TO DO ▼



+ Create issue

▼ **Backlog** (8 issues)

0 0 0

Create sprint

 IITMSE-2 Edit Catalogue

TO DO ▼



 IITMSE-3 Delete Catalogue

TO DO ▼



 IITMSE-5 Edit Inventory Item

TO DO ▼



 IITMSE-6 Delete Inventory Item

-

TO DO ▼



IITMSE Sprint 1



9 days remaining

Complete sprint



PP



GROUP BY

None ▾

Insights

TO DO 2 ISSUES

Add Catalogue



IITMSE-1

Add Inventory Item



IITMSE-4

IN PROGRESS

DONE ✓

Standup / Daily Scrum Meeting

- Daily meeting which involves - development team, scrum owner, product manager
- Each member answers three questions -
 - What did I work on yesterday?
 - What am I working on today?
 - What issues are blocking me?



Sprint Review

- Team demonstrates what they have completed
- Move things from To-Do, In Progress to Done



Sprint Retrospective

- Evaluate the last sprint - team dynamics, processes, tools etc.
- User stories/tasks that went well/didn't go well
- Create and implement a plan



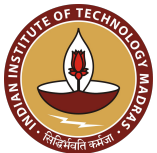
Summary - Activities in Scrum

- Sprint
- Scrum Team
- Sprint Planning
- Daily Scrum Meeting
- Sprint Review
- Sprint Retrospective



Project Scheduling in Agile

- Key indicator of progress - user stories implementation
- Project estimation - Count the number of user stories completed per iteration/sprint



▼ IITMSE Sprint 1 2 Aug – 16 Aug (2 issues)

0 0 0

Complete sprint



 IITMSE-1 Add Catalogue

TO DO ▼



 IITMSE-4 Add Inventory Item

TO DO ▼



+ Create issue

▼ Backlog (8 issues)

0 0 0

Create sprint

 IITMSE-2 Edit Catalogue

TO DO ▼



 IITMSE-3 Delete Catalogue

TO DO ▼



 IITMSE-5 Edit Inventory Item

TO DO ▼



 IITMSE-6 Delete Inventory Item

-

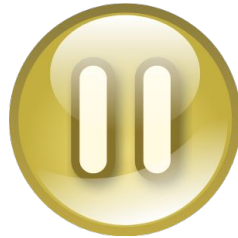
TO DO ▼



Reflection Spot

Project estimation - Count the number of user stories completed per iteration/sprint.

What can go wrong if we follow this approach for estimation?



Please pause the video and written down your responses



Points, Velocity

- Not all user stories require the same effort
- Rate each user story on a scale -
 - 1 - straightforward stories
 - 2 - medium stories
 - 3 - very complex stories
- Velocity - number of points per iteration/sprint



▼ IITMSE Sprint 1 2 Aug – 16 Aug (2 issues)

0 0 0

Complete sprint



 IITMSE-1 Add Catalogue

2 Points

TO DO ▼



 IITMSE-4 Add Inventory Item

2 Points

TO DO ▼



+ Create issue

▼ Backlog (8 issues)

0 0 0

Create sprint

 IITMSE-2 Edit Catalogue

1 Point

TO DO ▼



 IITMSE-3 Delete Catalogue

1 Point

TO DO ▼



 IITMSE-5 Edit Inventory Item

1 Point

TO DO ▼



 IITMSE-6 Delete Inventory Item

1 Point

-

TO DO ▼



Project Scheduling - Plan-and-Document vs Agile

- Plan and Document
 - Early in the project
 - Breakdown projects into tasks
 - Create Gantt charts, milestones
- Agile
 - User stories, Points, Velocity

