SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20031

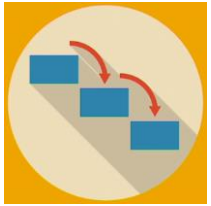
Computing Technology Design Project

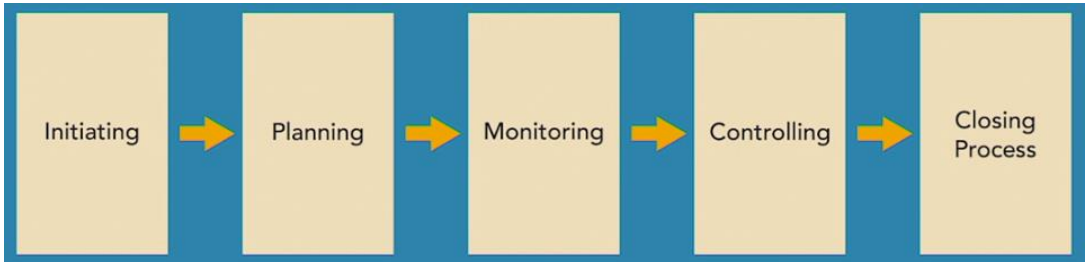
Week 02 - Project Planning and Task Management with Jira

(A)Project Planning

Traditional vs. Agile Project Management

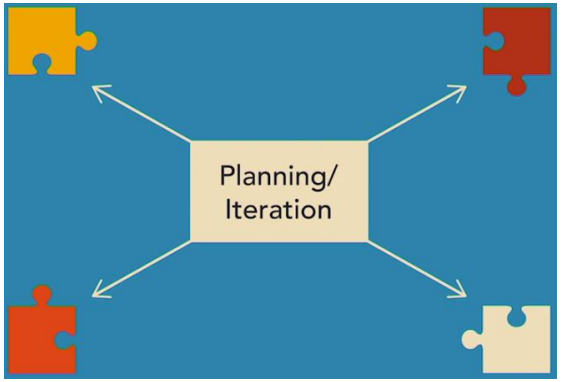
- **Traditional (waterfall)**:
  - works well when goals are clearly defined
  - complete one step then move to the next
- **Suitable** projects:
  - clear solution
  - simplicity
  - low risk
  - familiar technology
  - experienced resources

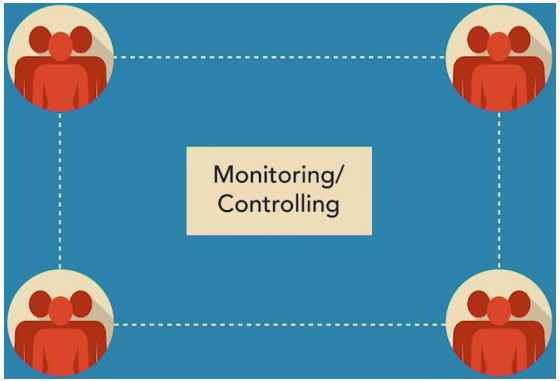




Traditional vs. Agile Project Management

- **Agile approach:**
  - iterations produce deliverables at regular intervals
  - value delivered sooner
  - more customer involvement
  - small independent teams
- **Suitable projects:**
  - unclear solution
    (most today's projects!)





PM process (basic)

1. Choose an PM approach
2. Identify stakeholders
3. Identify project goal
4. Identify project objectives
5. Identify project deliverables and success criteria
6. Prepare project scope statement
7. Create risk management plan
8. Create a risk chart
9. Learn to use milestones

Identify Stakeholders

- **Customer**
  - person or group with a problem to solve
  - funds the project
  - informs what needs to be done
  - approves deliverables
- Project **sponsor**
  - wants project to succeed
  - prioritise objectives
  - talk to stakeholders
  - suggest improvements
- Functional **managers**:
  - run departments to achieve department goals
  - manage team members
- **Team** members
  - jobs depend on performance in the projects
- **Department**:
  - invested and affected by project outcomes

Identify the Project Goal

- Defines the **end result**
  - Solves the problem, or
  - Takes advantage of opportunity
- **Problem statement**:
  - clearly defines the problem or opportunity
  - E.g. (hospital schedule system - HSS)
    "Hospital resources aren't being utilised efficiently, because scheduling doesn't ensure that the necessary equipment, staff and facilities are available. Fund is available to address scheduling issues."
- **Project goal**: e.g. (hospital schedule system - HSS):
  - "The project **will deliver** scheduling improvements **so** hospital resources can be scheduled efficiently. The project will take advantage of funding available for productivity and technology enhancements"

Identify the Project Objectives

- Objectives specify what needs to achieve to support organisational goal
- **Business** objectives
  - increase market share to 25%
  - provide world-class health care
- **Financial** objectives
  - increase revenue by 15%
  - cut costs by 10%
- **Quality** objectives
  - decrease staph infections by 80%
  - decrease readmittance by 75%
- **Technical** objectives: similar to technical specifications for equipment
  - introduce mobile equipment for hospital, ambulance and emergency site
- **Performance** objectives:
  - finish project before grant expires

- **S**pecific
- **M**easurable
- **A**chievable
- **R**ealistic
- **T**ime-related

Identify Project Deliverables & Success Criteria

- Deliverables: expected results
  - help define project scope
  - help measure progress
  - tangible: e.g. new product, service
  - intangible (abstract) deliverables: e.g. improved customer relationship
- Identify deliverables:
  - identify end deliverable
    - e.g. a new product/service launched
  - identify intermediate deliverables
    - achievable between status reports
    - e.g. sign a contract with a vendor

- Identify clear, quantifiable success criteria
  - signed contracts, certificate of occupancy, 4/5 survey rating

Prepare a Project Scope Statement

- Defines the boundary of project
  - specifies what is included (in) and what is excluded (out)
- Benefits:
  - help avoid scope creep (unexpected deliverables)
  - remind stakeholders of what was agreed upon
  - track for change management
- Scope statement: defines the scope
  - documents all the elements defined so far (goal, objectives, deliverables, success criteria, assumptions, risks, constraints)
  - boundary: what is within scope, out of scope

Scope example: HSS

Hospital Scheduling Project In-Scope

- Redesign processes for scheduling staff, equipment, and facilities
- Deliver new scheduling system and processes
- Document processes and system use
- Train staff on processes and system

Hospital Scheduling Project Out of Scope

- Update work shift scheduling system
- Scheduling resident rooms

Create a Risk Management Plan

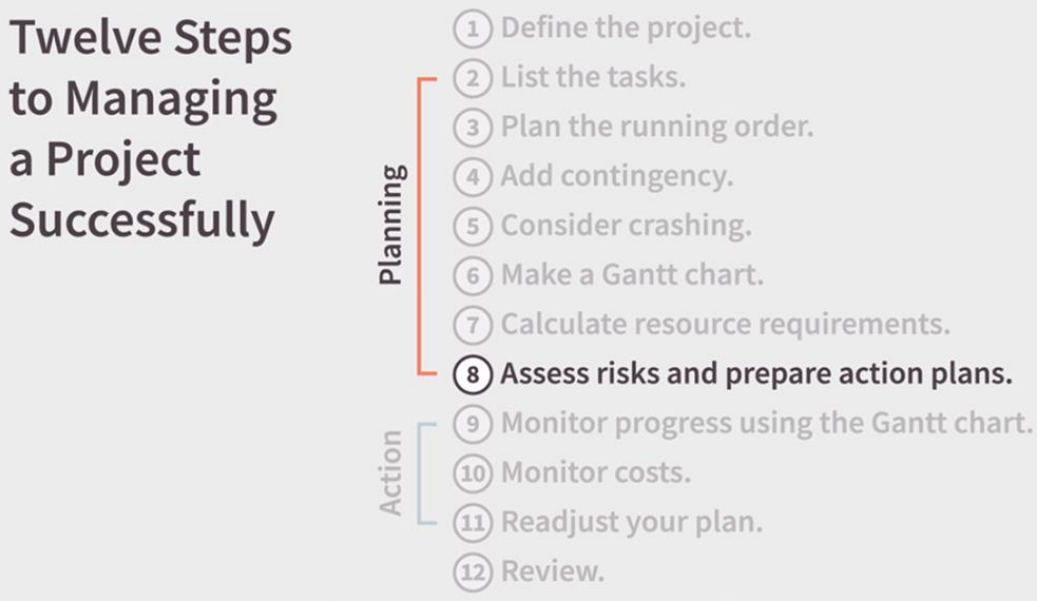
- Document the **relevant risks** and how we plan to **handle** them
- **Evaluate** risks:
  - probability or likelihood (1-5)
  - severity of impact (1-5)
  - risk score = probability x impact
- Rank risks in descending order of score
  - unlikely and not serious: 1 x 1 = 1
  - **medium** likely and serious: 3 x 3 = 9
  - very likely an serious: 5 x 5 = 25
- **Plan** for risks that are **medium or higher (i.e. score >= 9)**
  - response action to handle when a risk occurs
- Response **actions**: should reflect the risk
  - accept: for low risks
  - avoid: e.g. change project scope to avoid
  - mitigate: reduce impact (e.g. performs feasibility study)
  - transfer: to someone else (e.g. insurance)
- **Risk log**: document risk evaluation, response and status
  - update regularly
  - risk scores may change over time

Risk log example: HSS

Project	Scheduling system project									
last updated	12/10/23									
			Rank from 1 lowest to 5 highest	Rank from 1 lowest to 5 highest						
Risk ID	**Description**	**Trigger**	**Probability**	**Impact**	**Score**	**Response**	**Response Strategy**	**Expected Results**	**Status**	**Notes**
1	None of the vendor systems can provide the functionality required in the timeframe and budget required		5	5	25					
2	Systems aren't compatible with hospital equipment		4	4	16					
3	Scheduling delays push expenditures past the grant expiration date		4	3	12					
4	Key resource is pulled onto a different project		3	3	9					
5	Stakeholders disagree on required functionality		1	1	1					

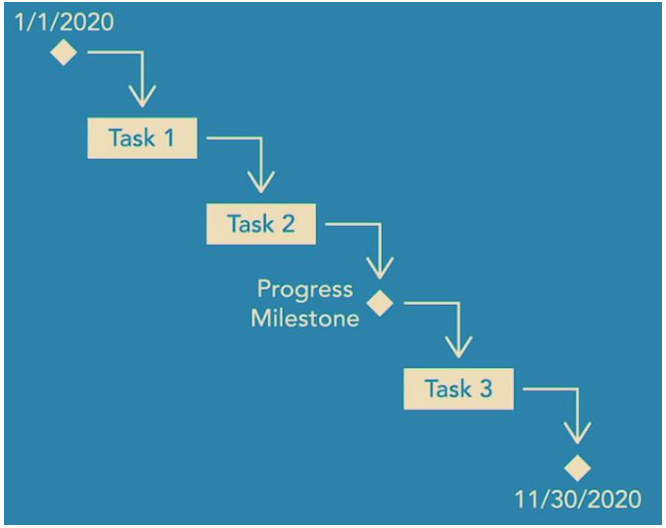
How to Create a Risk Chart

- Identify risks:
  - from previous projects
  - team brainstorming meeting
- Plan for risks with score > 5
- Actions:
  - prevention
  - protection
- Revise risk score
  - aim score <= 5
- Benefits:
  - informs stakeholders upfront
  - helps reduce risk



Learn to Use Milestones

- Milestone:
  - shows project progress
  - when key tasks have been completed
  - how much has been completed
  - when project is finished
- Helps easily reschedule the project
  - all tasks *dependent on* the milestone are rescheduled automatically
- Starts and ends project with a milestone
- Mark key progress points with a milestone:
  - key events, e.g:
    - wait for vendor delivery of equipment
    - plan approved
- Link tasks to milestones (before and after)



(B) Task management with Jira

Terms: basic

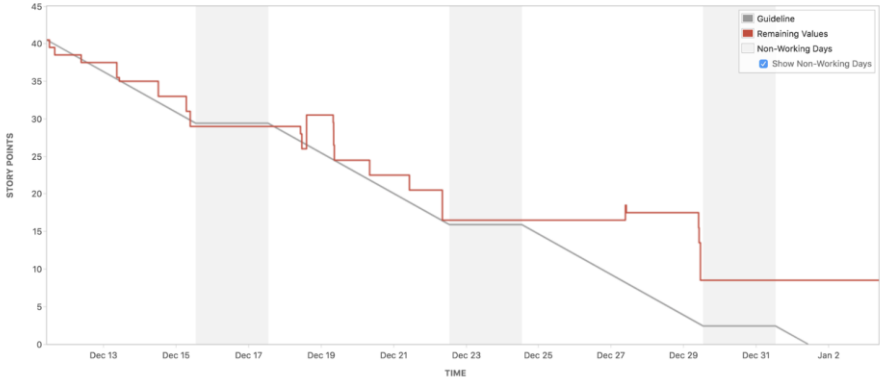
- **Scrum**
  - most popular frameworks for implementing agile: product is built in a series of small fixed-length iterations, called sprints, that can regularly be delivered
- **Backlog**:
  - prioritised list of ongoing tasks to complete, derived from requirements/roadmap
- User **stories**:
  - tasks or other work items => software features
  - a.k.a *issues* in Jira
  - As a {type of user}, I want {goal} so that I {receive benefit}
- **Epic**:
  - a body of work that can be broken down into user stories, having *flexible scope* based on the needs/requests of customers/end-users
  - usually consists of a set of sprints
  - can involve multiple teams and span multiple projects

Terms: Sprint (1)

- A set of user stories or other work items to be completed during a fixed time duration:
  - 1, 2 or 4 weeks
- Planning:
  - a meeting held at beginning of a sprint for entire team
  - duration: 2-hour for a 2-week sprint
  - discuss sprint goal
  - determine sprint backlog
    - refine stories (by creating substories)
    - estimate time for each story using story point
    - story point: relative work effort (fibonacci-like): 0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100
- Start a sprint:
  - start, end date
- Monitor sprint progress:
  - move items between to-do, in-progress and done columns

Terms: Sprint (2)

- Standup meetings:
  - one daily, starting meeting
  - duration: <= 15 minutes
  - usually informal (standup)
  - to update and check progress, resolve issues
  - questions for each member:
    - what did I do?
    - what will I work on today?
    - am i blocked by anything?
- View the Sprint report:
  - shows actual & estimated work effort left
    - Reports > Burndown Chart
  - x-axis: time
  - y-axis: story points (effort)



Terms: Sprint (3)

- Review meeting:
  - sprint review a.k.a sprint demo
  - typically on the last day of a sprint
  - duration: 2 hours for 2-week sprint
  - demonstrate what was completed in the sprint (a prototype or product increment) & update the product backlog
  - Questions to ask:
    - Did the team meet the sprint goal?
    - Any work added or removed during the sprint?
    - Did any work not get completed? If so, why?
  - Brainstorming on what's next
    - move unfinished stories to product backlog or a future sprint

Terms: Sprint (4)

- Retrospective meeting:
  - end of iteration
  - duration: 1.5 hours for 2-week sprint
  - team inspects itself for continuous improvement
  - Questions to ask:
    - What did we do well during the sprint?
    - What could we have done better?
    - What are we going to do better for next time?
- Complete the sprint
  - Click the “Complete sprint” button to close it

Terms: Epic

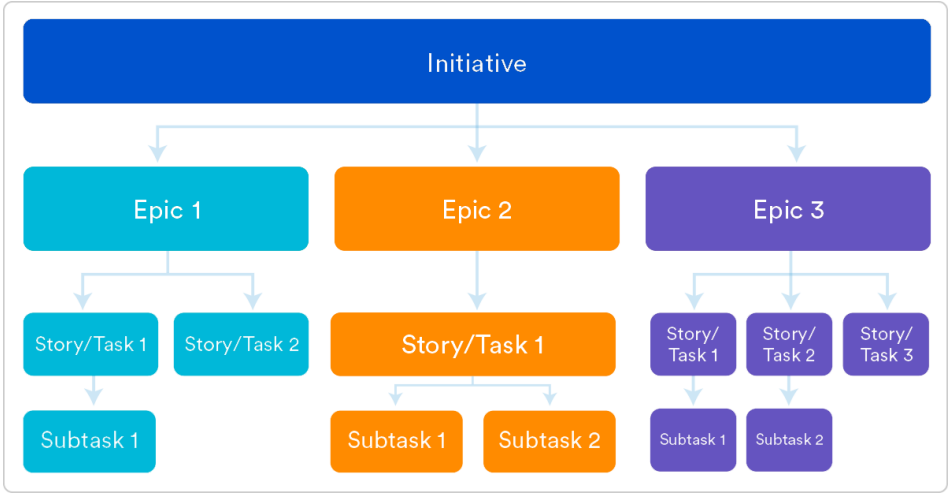
- A body of work that can be broken down into user stories, having flexible scope based on the needs/requests of customers/end-users
- usually consists of a set of sprints
- can involve multiple teams and span multiple projects
- Example: “March 2050 Space Tourism Launch”
  - various stories:
    - routine stories + stories for improving shuttle launch
    - e.g. space travel ticket purchase, rocket launch
  - multiple teams

Epic: March 2050 Launch		
Story: Update date range to include March 2050 Launch dates.	Story: Reduce load time for requested flight listings to < 0.45 seconds	Story: Promote Saturn Summer Sale on confirm page for First Class bookings.

Epic: March 2050 Launch		
Story: Keep fuel tanks PSI > 250 PPM on launch	Story: Reduce overall fuel consumption by 1%.	Story: Hire new propulsion engineer to replace Gary. #garygate2050

Terms: Theme, Initiative, Epic, Story (url)

- Product **road map**
  - **Theme**: an organisational goal
    - **Initiative**: a component of goal
      - **Epic**:
        - **Story**



Additional Sprint management with Jira

- URL: https://www.atlassian.com/agile/tutorials/sprints
- Optimise sprint with automation
  - send weekly Slack message on open issues
  - assign unfinished stories to the next sprint
  - move in-progress stories of an empty sprint to the next active sprint

Sprint Planning with Confluence and Jira

- **Jira**:
  - creates and manages the ***actual*** project plan (based on Agile)
- **Confluence**:
  - provides a documentation space for the project
  - provides templates for documenting the Sprint's activities (meetings, risks, etc.)
- Create Jira Project
  - can automatically create (and link) a new Confluence space
- Create a Scrum project with sample data:
  - read this
  - timeline: change Timeline setting to timebox each issue on the timeline
- Link Confluence space to Jira project
  - read this tutorial

Tutorial & Workshop

See Canvas.