1 Introduction

Software Development: The Goal is to make money by meeting need and requirements.

What is a Project?

- It has a specific target (product, result)
- It is Temporary - has an end.

What is Project Management?

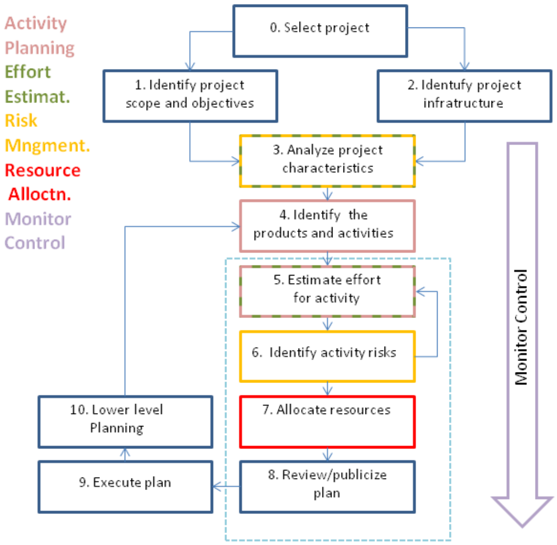
- Planning
- Organizing
- Staffing
- Directing
- Monitoring
- Controlling
- Innovating
- Representing

POSDMCIR

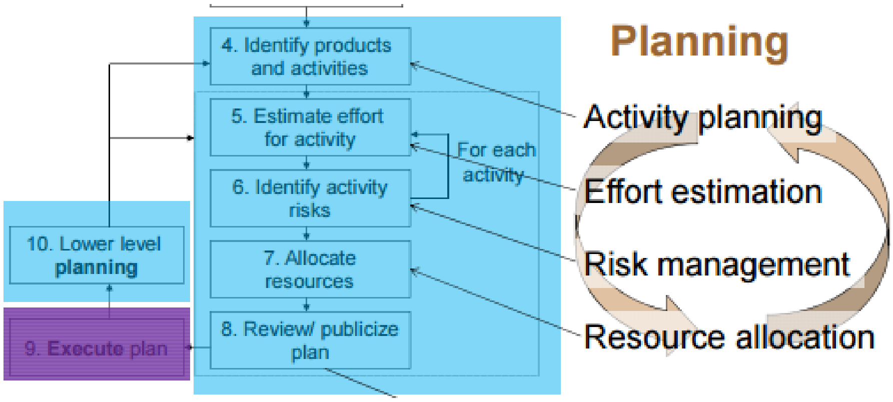
SW overruns the budget by 30% on average.
1/6 IT Projects ->Black Swan

SPM Areas

- Activity Planning
- Effort Estimation
- Resource Allocation
- Risk Management
- Monitor and Control Execution


Activity
Planning
Effort
Estimat.
Risk
Mngment.
Resource
Alloctn.
Monitor
Control
0. Select project
1. Identify project scope and objectives
2. Identufy project infratructure
3. Analyze project characteristics
4. Identify the products and activities
5. Estimate effort for activity
6. Identify activity risks
7. Allocate resources
8. Review/publicize plan
9. Execute plan
10. Lower level Planning
Monitor Control

The Planning is iterative and is gradually getting more *detailed*.



2 Föreläsning 1: Activity Planning

To reach a target:
- What should be done?
- In which order?
Activity Planning identifies *products* and *activities*.

Def. Activity:

- has start/end-points
- resource requirements can be forecasted
- a duration
- might be dependent on other activities

There are different approaches to identifying products and activities:

- Activity-based approach:
  - Work breakdown Structure (WBS)
  - Bottom-up: One adds up smaller tasks to get an overall estimate of effort, on divides and then puts than activities "back like a puzzle".
  - Top-down: One breaks down the project into smaller tasks, starts from the top both when it comes to identifying smaller tasks and working on them.
- Product-based approach:
  - Product Breakdown Structure: lists deliveries
  - Identify activities
  - Identify order
- Hybrid approach: iterative approach
  - Identify deliverables
  - Identify products
  - Identify activities

Critical Path:
A *path* where all activities have zero float.
- Any delay of a critical path delays he entire project.
- There are sub-critical paths which are impacted by: delays, changes due to resource allocation and estimation.

2.1 Föreläsning 2: Effort Estimation

Effort or Cost?

Effort = work required in project.
Effort during a project has a cost.

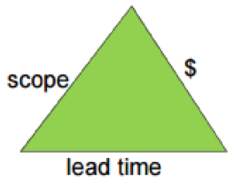


Figure 1: Man vill att triangeln ska vara liksidig = balanced project

- Successful projects deliver:
  - The agreed scope and quality
  - On time
  - Within budget

Problems with Estimating:

- Difficult to find evidence which supports your estimation - surety
- Changing technologies - must teach staff
- Projects differ from each other; one might not be able to compare with previous experience
- Political pressure: pressure from managers or investors to reduce cost
  ->bad quality

Over-estimate: *Parkinson's Law*
"Work expands to fill the time available"
->often leads to prolonging the project

Under-estimate: *Brook's Law*
"Putting more people on a late project makes it later"
->often leads to delays and lower quality

What one should take into account when estimating effort:

- People - skill, experience
- Requirements from clients might change
- Complexity of design and technology decisions

There are different approaches to estimate effort:

- Bottom-up
- Top-Down
- Code-oriented - estimate LOC
- Expert Judgment - "Guestimates"
- Analogy - Case-based, compare with other projects/principles
- Parametric or algorithm models - COCOMO, function points

GQM Framework

1. Define *goals*
2. Define *questions* that determine if goals are met:
- Refine goals
- Learn about progress towards goals
3. Define *metrics* that measure each question and determines if goal is actually achieved

3 Föreläsning 3: Resource Allocation

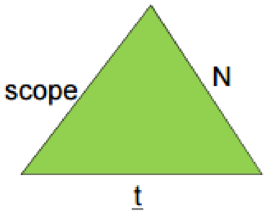


Figure 2: One should try to optimize the use of resources withing the budget ->(cost) and scope = time

What to consider:

- available resource types
- amount of tasks
- resources required per task - Resource Requirement List
- Time required for a resource needed for a task
- create resource histogram

Resources include for example:

- Labor
- Equipment
- Materials
- Space

Resource Requirement List:

STAGE	ACTIVITY	RESOURCE TYPE	mDAYS	QUANTITY

DON'TS: Overplan or Underplan (underplanera och överplanera, låter retarderat på engelska)

Resource Histogram

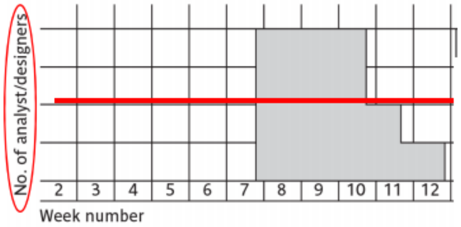


Figure 3: (Red line = max available resources) One can see that resource smoothing is needed

Resource Smoothing change resource allocation due to: - Limited number of resources
- Resource does not have the required competence
- Keep staff busy - "Brook's Law"
- Constant number of staff

Resource clashes is when the same resource is needed in several areas.

- can be solved by:
  - Delaying activities
  - Moving resource from non-critical activity
  - Add resource to task ->increase cost

4 Föreläsning 4

4.1 Risk Management

External Risks:
- Requirements change more than expected
- Expected input/deliverables are delayed
- Technology surprises
- Tools doesn't work as expected

Internal Risks:
- Staff turn-around
- Effort estimation is wrong
- Quality is not good enough at the end

One should make a *risk assessment*
->team focus on progress rather than problems

A *risk* consists of a cause, an effect and the probability of it *occurring*.

Risk Identification approaches:

- Brainstorming
- Checklists - often based on experience
- Casual mapping - find possible "risky chains" of cause and effect
- Probability matrix - Impact/Probability, a line is drawn where all activities should be under
- Decision tree - decisions are based on probability of cost

DON'TS:
- Don't add a safe zone (Parkinsons's and Brook's Laws)
- Identify too many risks ->spend time just investigating

"Damage" means different things to different stakeholders:

- Customers: budget overruns and delays
- Users: wrong functionality, weak product quality, usability problems
- Developers: poor-quality software architecture & design =>hard to maintain

Probability/cost of damage are also risky estimates
->mitigate by "buying information" – prototype - spike

Different stages of a risk assessment:

- Identification
  - Use checklists or brainstorming, compare with experience to come up with possible risks
- Analysis
  - Use performance or cost models to analyze the possible affect of an identified risk
- Prioritization
  - Make a risk-exposure analysis or a cost analysis to prioritize the different risks
- Planning
  - Prepare to address risks
- Resolution
  - Find solutions for risks, there are 5 alternatives: **AAMRT**
  Accept, Avoid completely, Mitigation(precautions e.g. back-ups), reduce impact and transfer(outsource problem to s/o else)

- Monitor
  - Track process and look for risks to avoid them

PERT - Program Evaluation Review Technique
PERT is a statical tool for analyzing completion time.

- Estimates:
  - Most likely time (m)
  - Optimistic time (o)
  - Pessimistic time (p)

Expected time = (o + p + 4m)/6
standard deviation = (p - o)/6

- Critical Chain approach:
  1. Ask for two estimations:
  - Most likely duration: 50% chance of meeting this estimation
  - Comfort zone: 90% chance of meeting
  this estimation
  2. Schedule activities using most likely, start activities on latest start dates
  - Reduce risk of loosing staff
  - Optimize time usage

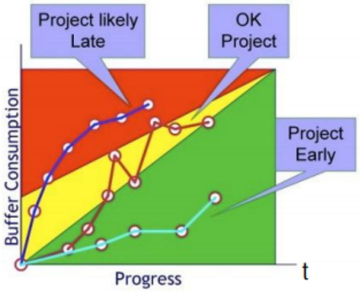


Figure 4: An example of a fever chart visualizing a Critical chain estimation

<u>While executing and monitoring CC plans consider:</u>

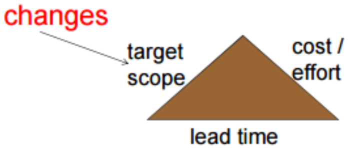
- Focus the efforts - NO MULTI-TASKING
  - No chain of tasks starts earlier than planned, but once it starts it should be completed as soon as possible
  - Tasks in chains starts in the planned order

Risk Management for Application Development Projects

- Risks are managed continuously during the development
  - Transparency with team members and clients
  - If too much i unknown ->"spike"
  - Mention hindrances at daily meetings

4.2 Agile Development: Scrum, XP, Kanban

Agile focuses on:
- ROI=return of investment, deliver business value
- Quickly delivering working code



Comparing with traditional projects agile projects "knows" less at the beginning of the project.
- It is gradually getting more detailed
- Self-governing teams
- Continuous feedback
- Same activities as a traditional but different sizing and sequence

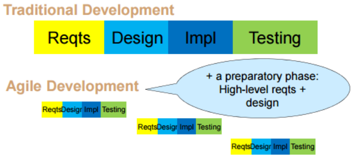

Traditional Development
Reqts
Design
Impl
Testing
Agile Development
+ a preparatory phase: High-level reqts + design
Reqts Design Impl Testing
Reqts Design Impl Testing
Reqts Design Impl Testing

Figure 5: Time-Box: fixed deadline for deliverables

5 Föreläsning 5

5.1 Monitor & Control Execution

The final goals for a successful monitor and control over execution:
- Scope: well-implemented product or other result
- Budget
- Timeliness
Examples of tools to monitor and control: **Gantt Chart**, **Slip Chart**

Principles of Monitor & Control

- Project can be *behind in schedule* but *under budget*
- Project can be *on time* but *over budget*
- Focus on monitoring *based on risks*

One need to monitor both cost and achievements.

Monitor approaches can be event or time driven:
- Collecting data
- Reporting

Control; getting back on schedule
- Renegotiate deadlines (cost, scope, time)
- Shorten critical path
- Reconsider activity dependencies

5.2 Quality in Project

There are 4 types of downfalls which are in the way for a successful project:

- Delays
- Inadequate functionality
- Inadequate quality
- Cost overruns

Quality Management is trying to avoid *inadequate functionality* and *inadequate quality*.

Quality is the ability to *satisfy the needs* and *expectations* of the customer.
One should *balance* cost and quality.
Consider the ISO SW qualities: FRUEMP.

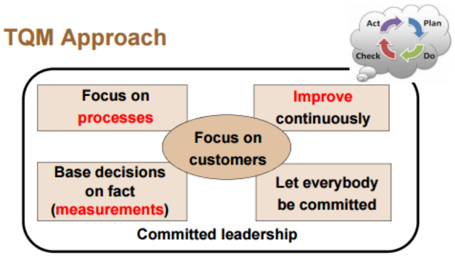


Figure 6: TQM - Total Quality Management

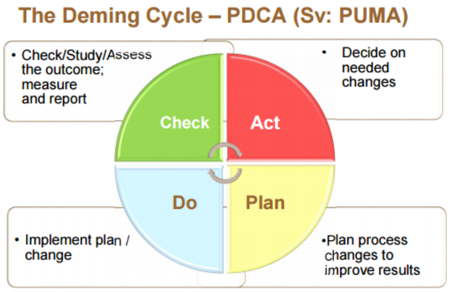

The Deming Cycle – PDCA (Sv: PUMA)
• Check/Study/Assess the outcome; measure and report
• Decide on needed changes
Check
Act
Do
Plan
• Implement plan / change
•Plan process changes to improve results

Figure 7: The Deming Cycle - **PDCA**

The general steps of SPI: Software Project improvements
1. Evaluation of current process (Check)
2. Planning to improve (Act & Plan)
3. Implement improvements or changes (Do)
4. Evaluate effects (Check)
SPI is iterative, so the steps continues throughout the process.

The different SPI approaches:

- Prescriptive: Compare current practice with best practice
  Ex: SPICE, CMMI
- Inductive: Identify problems in current practice and fix
  - Lean Six Sigma, iFlap, Quality improvement algorithm, Retrospective analysis

6 Föreläsning 6: Managing People

Management throughout the project:
- Stakeholders (people-related goals)
- Uniform process for different units
- Motivation
- Estimates are people-dependent
- Risks are staff-related
- Allocate individuals
- Communication

It is important to have clear goals for team members to motivate.
Stress can be *reduced* by good Project Management.

Taylorism
+ Select the best people for the job
+ Instruct them with the best methods
+ Performance-related pay
- "Inflation" in pay
- Staff exhaustion

Hawthorne Effect
Experiments showed that staff performed better when:
- Improve lighting
- Clean work stations
- Relocating work stations
There are 2 types of people:
X: dislike work, avoid responsibility
Y: seek responsibility, work is natural

The Oldham-Hackman job characteristics

A job should be/include:

- Meaningful
  - Skill variety
  - Task identity: sense of ownership - Task significance
- Autonomy
  - Authority
  - Responsibility
- Feedback
  - Rewarding - constructive

7 Föreläsning 7: Portfolio Management

Return of Investment = NET PROFIT / TOTAL INVESTMENT
->*The time it takes to achieve the same amount of money that was invested in the process/project*

Types of Contracts

- Fixed price (items, requirements, delivery time)
- Time and Material contracts
  - fixed cost per unit of effort
- Fixed price per delivered unit