The failure rate for big software projects is downright scary. Depending on whose study you read, the ERP failure rate is anywhere between 50% and 75%. So going in management knows the project is likely to fail. Yet in the 1990s and early 21st century there were still companies willing to try their hand with big software and prove they were not like the others who failed so spectacularly. In spite of this optimism, many of these companies also failed. Even the US Defense Department failed.

The Alaska Department of Public Safety spent 11 years and $27.9 million trying to replace the state's aging criminal information database before the project was abandoned.

Australian Census website collapses despite millions spent on IT contracts

In mid-January 2016 the Nest 'smart' thermostat (owned by Google) was hit with a software glitch which left users, literally, out in the cold.

A software update went wrong, forcing the device's batteries to drain and leaving it unable to control temperature – so customers were unable to heat their homes or get hot water on one of the coldest weekends of the year so far.

The US National Grid Gas Company moved to a new ERP system from SAP in 2012 in an effort to streamline back-office processes. However the software was incorrectly implemented, resulting in problems like inaccurate wage payments and unpaid supplier bills. The cost of implementing and fixing the software means it has cost $945 million (£607 million), up from an original estimate of $383 million (£248 million), according to external auditors.

What's going on here!!!!

Q. What do you think are the problems?

- Software Development Processes (reqs, design, coding, testing, etc).

- Software process models (also known as **Lifecycle Models**) are general approaches for organizing these processes in a project. They prescribe:
  - When to start each process (like reqs, design)
  - What is criteria for transitioning between processes.

- They help the project manager and his or her team to decide:
  - What work should be done
  - Tasks, milestones, deliverables

Different types of projects will need to handled differently.

Using the same cookie-cutter approach to all projects will result in failure.

Project-1

- You are on your way to get your first contract from a real customer!

You have to create a well-documented, well-tested binary search program that the customer can use to search information he has in a data structure.

  - First, you have to draw up a plan of work and get approval
    - milestones and dates
    - a time-estimate
    - a cost-estimate

Example: binary search

- Characteristics
  - Well-understood
- Plan of work
  - Define interface (1/2 hr)
  - Develop code (1 hr)
  - Test code (2 hrs)
  - Finalize documentation
  - Time Estimate?
  - Cost Estimate?

EASY TO PLAN ACCURATELY!
Few risks in project!

waterfall approach

Project-2

– What if you are given a different task?

You are to create a well-documented, well-tested topological sort program that the customer can use to sort information he has in a data structure.

– First, you have to draw up a plan of work and get approval
  - milestones and dates
  - a time-estimate
  - a cost-estimate

Example: Topological sort

- Characteristics
  - What is topological sort?
  - Probably documented in some text
- Plan of work
  - Research! Find out and understand (1/2 day)
  - Play around with algorithm (1/2 day)
  - Define interface (1/2 hr)
  - Develop code (2 hrs)
  - Test (2 hrs)

Harder to plan.
Known ways to solve.
Not a risky project!
Prototype approach

Project-3

What if the new goal is to create a well-documented, well-tested library of different sort programs that the customer can use to sort information he has in a data structure.

- First, you have to draw up a plan of work and get approval
  - milestones and dates
  - a time-estimate
  - a cost-estimate
- How will you proceed?

Example: Library of sort algos

- Characteristics
  - Known sort algos
  - Obvious increments
- Plan of work
  - Phase 1
    - Develop sort 1 and release
  - Phase 2
    - Develop sort 2 and release
  - Phase 3
    - Develop sort 3 and release

Better to develop in an iterative fashion.

Project-4

Next – the goal is to create a well-documented, well-tested sort program that returns results within 1 millisecond of request even for peta bytes of data!

- First, you have to draw up a plan of work and get approval
  - milestones and dates
  - a time-estimate
  - a cost-estimate
- How will you proceed?

Example: Sorting algo that returns results within 1 millisecond of request!

- Characteristics
  - Very stringent requirements
  - Many unknowns, high risk
- Approach
  - Cycle 1
    - Develop an algo and collect some data
    - Scrap project??
  - Cycle 2
    - Analyze data and do some feasibility studies and propose some solutions
    - Scrap project??
  - Cycle 3
    - Develop a solution?

Better to develop in an spiral/iterative fashion.

Very Risky project. May need to be aborted.

Selection of an approach

- choose an approach based on the nature or type of the project.

- Some questions to ask…
  - Are the reqs defined clearly?
  - Is quality/reliability all important? (safety critical)
  - Does this project have stringent performance requirements?

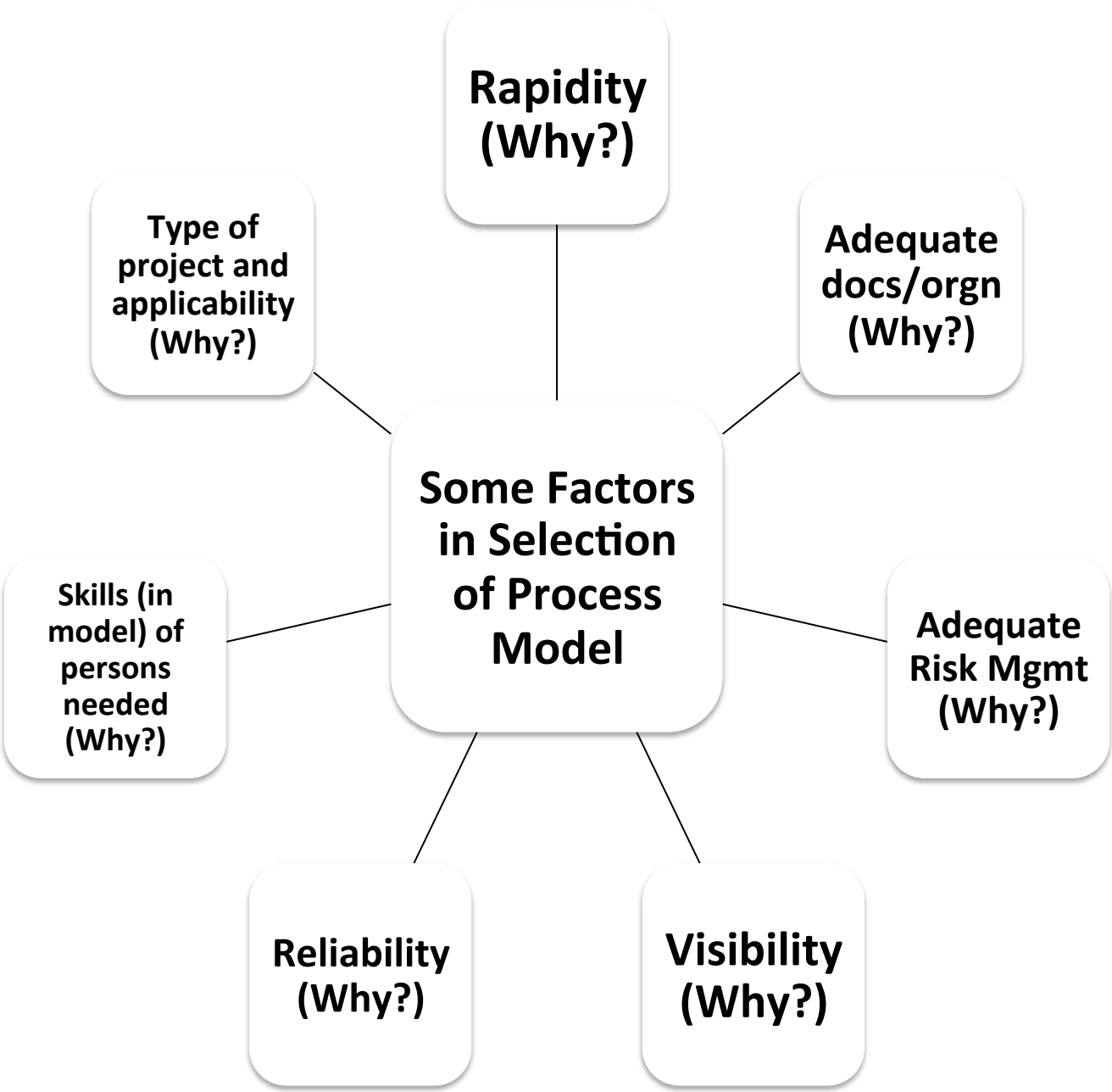
  - Is this being built on top of an existing architecture?
  - Have we done similar projects?

  - Is it important to develop rapidly?
  - what is the experience and skill-level of our teams?
  - ……

from a management point of view

How will the following concerns be addressed?

- Visibility (at any time, how much work is left?)
- Rapidity (time-to-market)
- Reliability (trapping errors)
- Risks-handling?


Rapidity (Why?)
Type of project and applicability (Why?)
Adequate docs/orgn (Why?)
Some Factors in Selection of Process Model
Skills (in model) of persons needed (Why?)
Adequate Risk Mgmt (Why?)
Reliability (Why?)
Visibility (Why?)

Remember the main goal

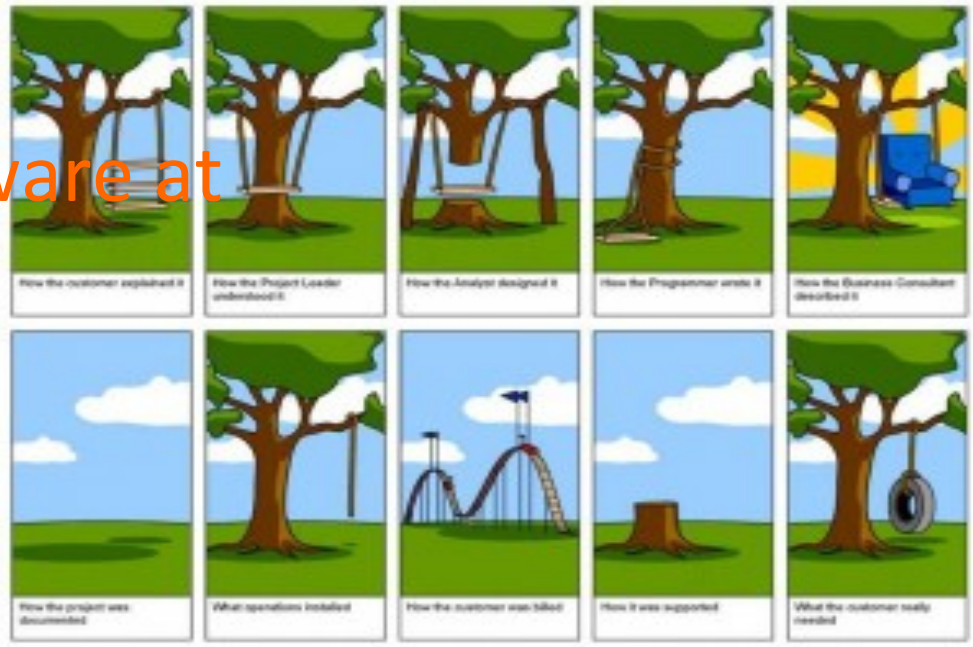
Software Engineering can be defined as methods and procedures for software development that can be used to deliver

(1) high QUALITY software at

(2) low COST and a

(3) small cycle TIME

(4) **CONSISTENTLY**



Why do you need SE?

- a birdhouse?
- a small shed?
- a house?
- a skyscraper?

Programmer

Software Engineer

- Are the issues the same?
- Are the skills needed the same?
- Are the processes the same?
- Did the paradigm shift happen?

Self Check

- What is a **Process Model**?
- What are some factors to consider when **choosing** a process model for a project?
- What is visibility, rapidity, reliability of a process model?