Who needs software?

- Most software is built in organizations for people with specific needs.
  - A **stakeholder** is anyone who has an interest (or stake) in the software being completed.
  - A **user** is someone who will need to use the software to perform tasks.
- Sometimes stakeholders will be users; but often the stakeholder will not use the software.
  - For example, a senior manager (like a CEO or CTO in a company) will usually have a stake in the software that is built (since it affects the bottom line), even if she won’t ever use it.
  - Stake:- one used as a support,
Who builds software?

Software is typically built by a team of software engineers, which includes:

- *Business analysts* or *requirements analysts* who talk to users and stakeholders, plan the behavior of software and write software requirements
- *Designers and architects* who plan the technical solution
- *Programmers* who write the code
- *Testers* who verify that the software meets its requirements and behaves as expected
Project Management

- The project manager plans and guides the software project
  - The project manager is responsible for identifying the users and stakeholders and determining their needs
  - The project manager coordinates the team, ensuring that each task has an appropriate software engineer assigned and that each engineer has sufficient knowledge to perform it
  - To do this well, the project manager must be familiar with every aspect of software engineering
Identifying Needs

- The project manager drives the scope of the project.
  - The project manager should identify and talk to the main stakeholder
  - The effective way to show stakeholders that their needs are understood and that those specific needs will be addressed is with a *vision and scope document*
A typical vision and scope document follows an outline like this one:

1. Problem Statement
   a) Project background
   b) Stakeholders
   c) Users
   d) Risks
   e) Assumptions
2. Vision of the Solution
   a) Vision statement
   b) List of features
The project plan defines the work that will be done on the project and who will do it. It consists of:

- A statement of work that describes all work products that will be produced and a list of people who will perform that work
- A resource list that contains a list of all resources that will be needed for the product and their availability
- A work breakdown structure and a set of estimates
- A project schedule
- A risk plan that identifies any risks that might be encountered and indicates how those risks would be handled should they occur
Resource List

- The project plan should contain a list of all resources that will be used on the project.
  - A resource is a person, hardware, room or anything else that is necessary for the project but limited in its availability.
  - The resource list should give each resource a name, a brief one-line description, and list the availability and cost (if applicable) of the resource.
The project plan should also include estimates and a project schedule:

- A work breakdown structure (WBS) is defined. This is a list of tasks which, if performed, will generate all of the work products needed to build the software.
- An estimate of the effort required for each task in the WBS is generated.
- A project schedule is created by assigning resources and determining the calendar time required for each task.

*Estimates and project schedules will be discussed in detail in later slides.*
A *risk plan* is a list of all risks that threaten the project, along with a plan to mitigate some or all of those risks.

The project manager selects team members to participate in a risk planning session:

- The team members brainstorm potential risks
- The probability and impact of each risk is estimated
- A risk plan is constructed
PM Process Phases

- There are three broad phases
  - Planning
  - Monitoring and control
  - Termination analysis

- Planning is a key activity that produces a plan, which forms the basis of monitoring
Planning

- Done before project begins
- Key tasks
  - Cost and schedule estimation
  - Staffing
  - Monitoring and risk mgmt plans
  - Quality assurance plans
  - Etc.
Monitoring and control

- Lasts for the duration of the project and covers the development process
  - Monitors all key parameters like cost, schedule, risks
  - Takes corrective actions when needed
  - Needs information on the dev process – provided by metrics
Termination Analysis

- Termination analysis is performed when the development process is over.
- Basic purpose: to analyze the perfection of the process, and identify lessons learned.
- Also called post-mortem analysis.
Relationship with Dev Process
Timeboxing Model – Basics

- Development is done iteratively in fixed duration time boxes
- Each time box divided in fixed stages
- Each stage performs a clearly defined task that can be done independently
- Each stage approximately equal in duration
- There is a dedicated team for each stage
- When one stage team finishes, it hands over the project to the next team
Example

- An iteration with three stages – Analysis, Build, Deploy
  - These stages are appx equal in many situations
  - Can adjust duration by adjusting the team size for each stage
- Have separate teams for A, B, and D
Pipelined Execution

- AT starts executing it-1
- AT finishes, hands over it-1 to BT, starts executing it-2
- AT finishes it-2, hands over to BT; BT finishes it-1, hands over to DT; AT starts it-3, BT starts it-2 (and DT, it-1)
- ...

...
Timeboxing Execution

TB1
Requirements | Build | Deploy

TB2
Requirements | Build | Deploy

TB3
Requirements | Build | Deploy

TB4
Requirements | Build | Deploy

Software
Work Allocation of Teams

<table>
<thead>
<tr>
<th>Requirements Team</th>
<th>Requirements Analysis for TB1</th>
<th>Requirements Analysis for TB2</th>
<th>Requirements Analysis for TB3</th>
<th>Requirements Analysis for TB4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Team</td>
<td>Build for TB1</td>
<td>Build for TB2</td>
<td>Build for TB3</td>
<td>Build for TB4</td>
</tr>
<tr>
<td>Deployment Team</td>
<td>Deployment for TB1</td>
<td>Deployment for TB2</td>
<td>Deployment for TB3</td>
<td></td>
</tr>
</tbody>
</table>
Time boxing

- Advantages: Shortened delivery times,
- Disadvantages: Larger teams, project mgmt is harder,
### Summary – Timeboxing

<table>
<thead>
<tr>
<th>Strength</th>
<th>Weakness</th>
<th>Types of Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>All benefits of iterative Planning for iterations somewhat easier</td>
<td>PM becomes more complex</td>
<td>Where very short delivery times are very important.</td>
</tr>
<tr>
<td>Very short delivery times</td>
<td>Team size is larger</td>
<td></td>
</tr>
<tr>
<td></td>
<td>More complication leads to losses.</td>
<td></td>
</tr>
</tbody>
</table>
Using Process Model in a Project

- Model to be used should be selected based on the nature of the problem
- Example: Build a small auction system for a University, tight schedule, some core req, customer time only in start,...
- Suitable model: Iterative delivery – do req in 1st iter; and two rounds of delivery; minimizes risk,...
Using Process Models..

- Example: Highly competitive product; requirement change rapidly; outsourcing is desired for reducing cost,…

- Model: iterative may not deliver rapidly enough; time boxing best suited.