## Software Process

- Software life-cycle model: high level, strategic decisions around the overall mode of the development lifecycle (e.g. try to define requirements upfront or opt instead to evolve the requirements over time).
- Software Process defines all of the details associated with the process:
  - Software life-cycle model
  - Tools to use
  - Individuals building software
  - Individual process tasks.

1

1

## Why Model software processes?

- To provide a common understanding
  - To whom?
    - Everyone…
- To locate and remove
  - Inconsistencies
  - Redundancies
  - Omissions
- To understand the development goals and provide early evaluation
- To assist the development team to understand any special situation

2

2

## Software Development Life Cycle

"You've got to be very careful if you don't know where you're going, because you might not get there."

*Yogi Berra*


3

3

## Life-Cycle Models

- Build-and-fix model
- Waterfall model
- Rapid prototyping model
- Spiral model
- Prototyping Model
- Phased Development Model
  - incremental development model
  - iterative development model
- Formal Systems Development
- Agile models (model or method)
  - Extreme programming

We will talk about these in more detail over the next few lectures

See – Ian Sommerville, Software Engineering, any edition for good chapter on process models

4

4

## What is the best process?

- **There is no "best" process** or process model
- There is a most appropriate process for any given context
  - What is context?
  - These are the circumstances that a software development effort are faced with
  - The generic models presented earlier are appropriate only in a generalised type of way
  - No two software development contexts are identical
  - Some degree of *synthesis | tailoring | adaptation* of a generic model (or models) is required to address the contextual factors

5

5

## Recall: Complexity Issues


6

6

## Software Process

- Defined as "*A set of activities, methods, practices and transformations that people use to develop and maintain software and the associated products (e.g. project plans, design documents, code, test cases and user manuals)*"
- A software process has four distinct roles:
  1. To provide guidance as to the order of the activities to be undertaken
  2. To specify the artefacts that should be developed and when
  3. To direct the tasks of the development team
  4. To offer ways of monitoring and measuring a project's progress and outputs.

7

7

## Software Process Improvement

- SPI aims to understand the software process as it is used within an organisation and thus drive the implementation of changes to that process to achieve specific goals such as increasing development speed, achieving higher product quality or reducing costs.
- The reason for this focus on SPI is encapsulated in the belief that there is an intrinsic **link between the quality of the software process and the quality of the outputs emanating from that process**.
- Process improvement is about making things better – not about fire fighting or handling crises.
  - It is about stop blaming "someone" for problems or faults.
  - It is a way to look at how we can do our work better.

8

8

## Question: Is there a link between Process and Product quality?

**Improved software process**

**Improved software product**

9

9

## Process and Product quality

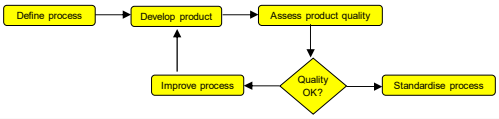
- The quality of a developed **product** is influenced by the quality of the production **process**.
- This is important in software development as some product quality attributes are hard to assess.
- However, there is a very complex and poorly understood relationship between software processes and product quality.

10

10

## Process-based quality

- There is a straightforward link between process and product in manufactured goods.
- More complex for software because:
  - The application of individual skills and experience is particularly important in software development;
  - External factors such as the novelty of an application or the need for an accelerated development schedule may impair product quality.
- Care must be taken not to impose inappropriate process standards - these could reduce rather than improve the product quality.




11

11

## Continuously Improve Processes

**Our aim here is to…**

Understand the software process as it is used within an organization and thus drive the implementation of changes to that process to achieve specific goals such as achieving higher product quality, reducing costs, etc.

*It is a widely accepted fact that the quality of a software product is largely determined by the quality of the process used to maintain and develop it* (Zahran, 1998)

*To improve your product, you must improve your process quality* (Humphrey 1995)

12

12

## Hallmarks of Our Industry

- It's an unfortunate truth that many projects are:
  - Late: deadlines are usually missed
  - Over budget: cost more than planned
  - Not fully functional: don't have all the promised features and functionalities
  - Defective: deployed with defects and need heavy re-work to remove them.
- *We do the same thing over and over again, and expect different results.*

13

## Common (Mis)conceptions about Software Process

- Investment in Process = More Cost
- Industry operates at such a low margin that it cannot afford processes.
- Processes increase overheads, resulting into more development and/or deployment time.
- Processes make an organization inefficient. Our time is consumed by documentation, rather than coding.

14

## Realities about Software Process

- Industry operates on low profits because it does not have efficient processes.
- Engineers do lot of re-work because of incapability of existing processes.
- If a process makes you do *unnecessary* work, then its not an efficient process.
- An effective process must reduce your response time, increase productivity and stabilize your internal systems.
- Continuous Software Engineering has started to automate some previously manual tasks.

15