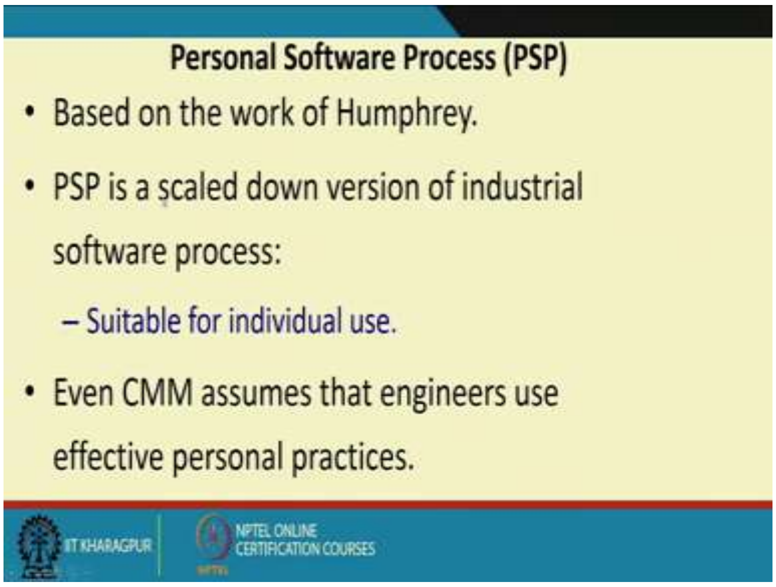
Software Project Management
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 56
Personal Software Process (PSP)

Welcome to this lecture, over the last couple of lectures we had discussed some basic concepts in Software Quality Management and we had looked at quality standards. The ISO 9001 and SEI CMM. We had also compared these two standards, which will give us an insight in as to which one is more beneficial to use for a specific organization.

(Refer Slide Time: 01:02)



Personal Software Process (PSP)

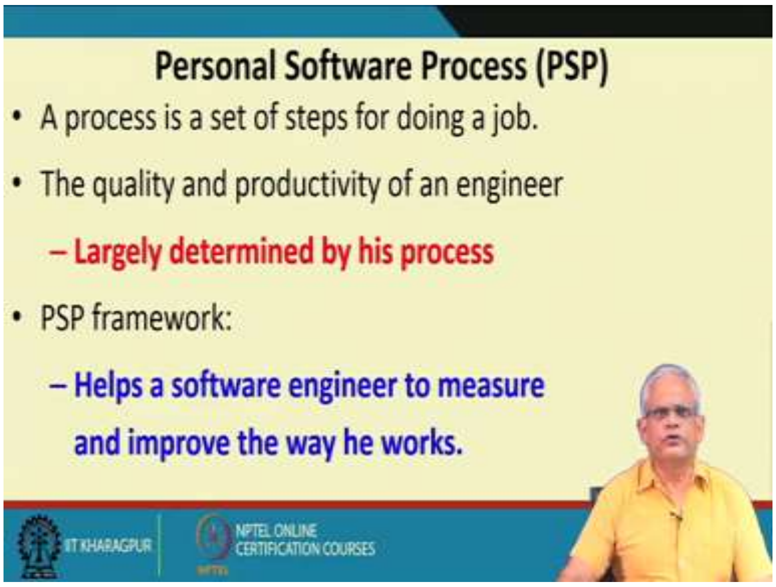
- Based on the work of Humphrey.
- PSP is a scaled down version of industrial software process:
  - Suitable for individual use.
- Even CMM assumes that engineers use effective personal practices.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

In this lecture we look at the Personal Software Process. The ISO 9001 and SEI CMM are applicable to organizations, these look at the process that is being followed at the organization and then they make recommendations on the quality issues of this process. But the need for a personal software process was identified because, every individual uses his own personal process. These are actually scaled down version of the industrial software process and are used by individuals, unlike the SEI CMM and the ISO 9000 which are used across an organization across all projects in an organization.

Here it focuses on the individuals working style or the individuals process, ISO 9000 and CMM they just make an assumption then individuals have effective personal practices. But then how do the individuals improve their personal practices starting with poor practices, how can they improve their practices over stages and that is the focus of the personal software process or PSP.

(Refer Slide Time: 02:49)



Personal Software Process (PSP)

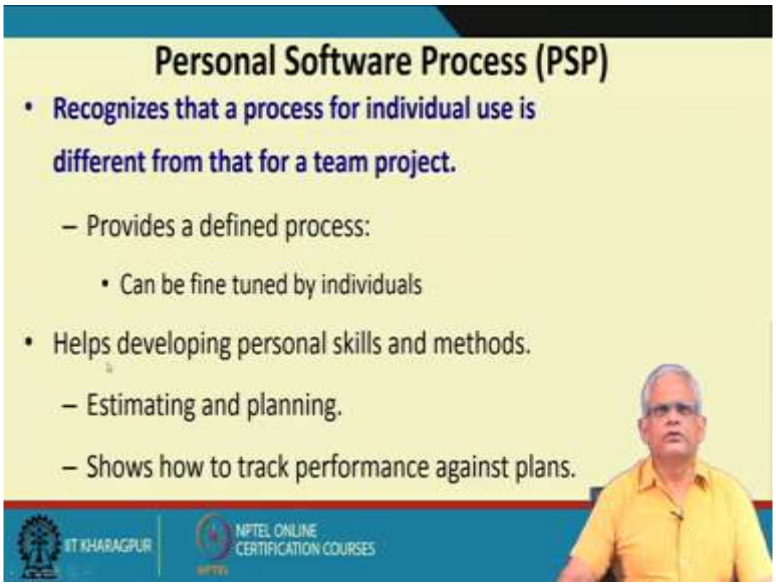
- A process is a set of steps for doing a job.
- The quality and productivity of an engineer
  - Largely determined by his process
- PSP framework:
  - Helps a software engineer to measure and improve the way he works.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

As we know a process is the set of steps that somebody uses for doing a job. For example, a developer is given the task of coding a module, how does he go about? Does he just understand the module requirement and then start coding, does he look at the module make an estimate of the code, the coding time and then really keep the time and then check whether he is according to the plan that he has made, does he have a coding process how to code.

So, these are the ones that are targeted by the PSP. The quality and productivity of a engineer is largely determined by his process, the PSP framework it helps a software engineer to measure and improve the way he works.

(Refer Slide Time: 04:08)



Personal Software Process (PSP)

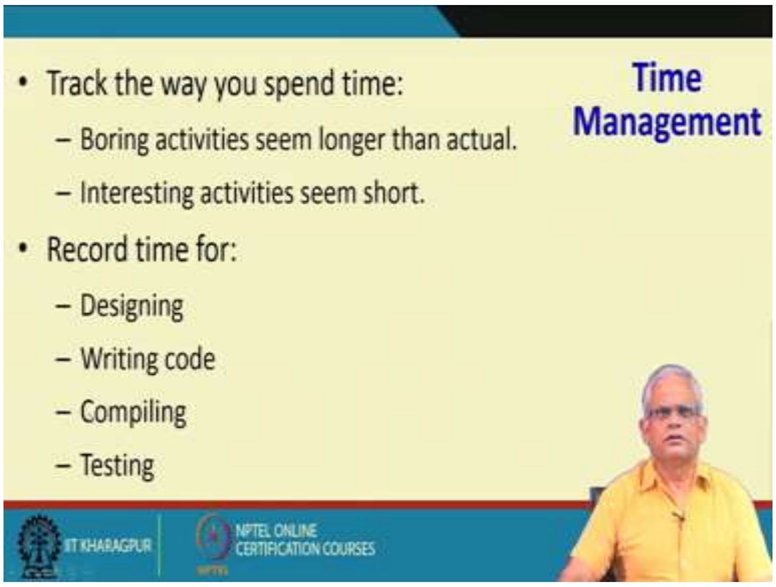
- **Recognizes that a process for individual use is different from that for a team project.**
  - Provides a defined process:
    - Can be fine tuned by individuals
- Helps developing personal skills and methods.
  - Estimating and planning.
  - Shows how to track performance against plans.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

The individuals they have their different styles of working and it provides a defined process for the individuals and of course the individuals can finetune and continually improve their own process. It helps individuals to incorporate certain personal skills and methods. For example, they can do estimation and planning for their own work the project manager does estimation and planning for the project.

But what about the developers do they do any estimation of their own work that is assigned to them. Do they plan out how to go about doing that work and do they really track the performance against the plan they made.

(Refer Slide Time: 05:13)



Time Management

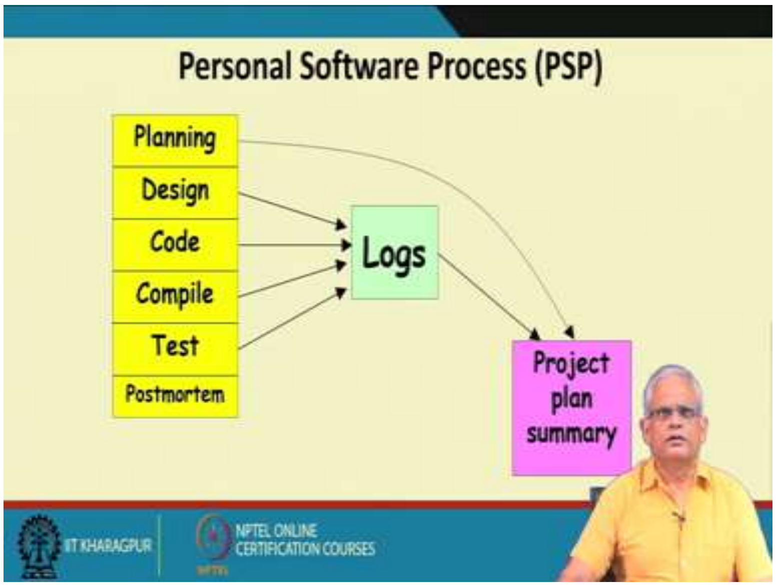
- Track the way you spend time:
  - Boring activities seem longer than actual.
  - Interesting activities seem short.
- Record time for:
  - Designing
  - Writing code
  - Compiling
  - Testing

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

One of the most basic issues that has been pointed out by the personal software process is the time management. Unless the time is measured it becomes very difficult or very erroneous if somebody tries to do a boring activity which he does not really enjoy, then it will appear to be much longer than it is. For example, somebody let us say does not like to design he likes to code and he is asked to design even if he spends just few hours, he may say that I have been doing the design activity for a long time and therefore the time needs to be measured.

Similarly, interesting activity seem very short, somebody who enjoys doing coding and he has coded for let us say eight straight hours and you ask him that how long did he code he will say that no I just spent very little time in coding, I just done it all without spending too much time. For this reason it is necessary to record the time for designing, writing code, for compiling and removing the syntax errors and then testing and removing the bugs.

(Refer Slide Time: 06:56)



Personal Software Process (PSP)

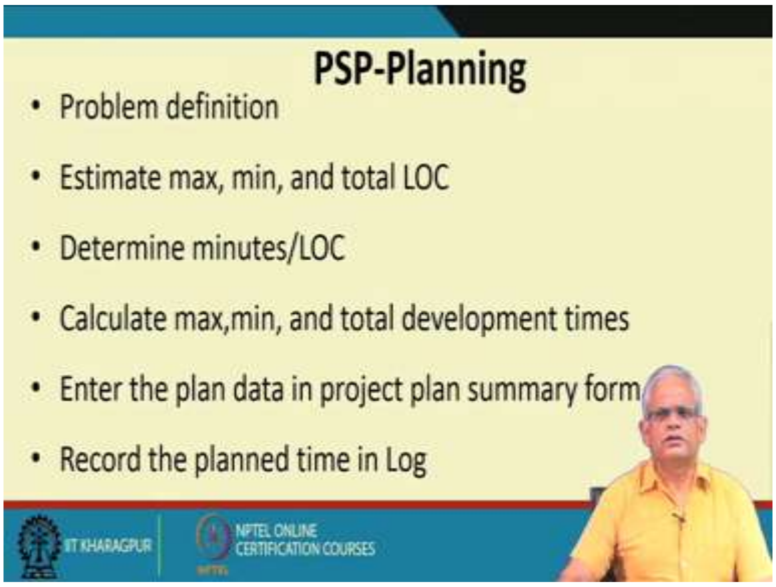
$$\begin{tikzcd}
\text{Planning} \arrow[rrd, bend left] & & \\
\text{Design} \arrow[r] & \text{Logs} \arrow[r] & \text{Project plan summary} \\
\text{Code} \arrow[ru] & & \\
\text{Compile} \arrow[ruu] & & \\
\text{Test} \arrow[ruuu] & & \\
\text{Postmortem} & &
\end{tikzcd}$$

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Plans have to be made by the individual if he is given a unit to code, he has to plan how is he going to do it has to write down the plan for the design must record the time taken in the log and the plan must be ok. That is compared later with the log for the coding how much time really it took compilation to remove syntax errors, how much time does it took test to remove the bugs, how much time does it took and these are all compared with the initial plan made and a project plan summary is developed.

And then a postmortem is used to identify what went right what did not go ok, how it needs to be improved and so on. So, that is the basic of the PSP.

(Refer Slide Time: 08:09)



PSP-Planning

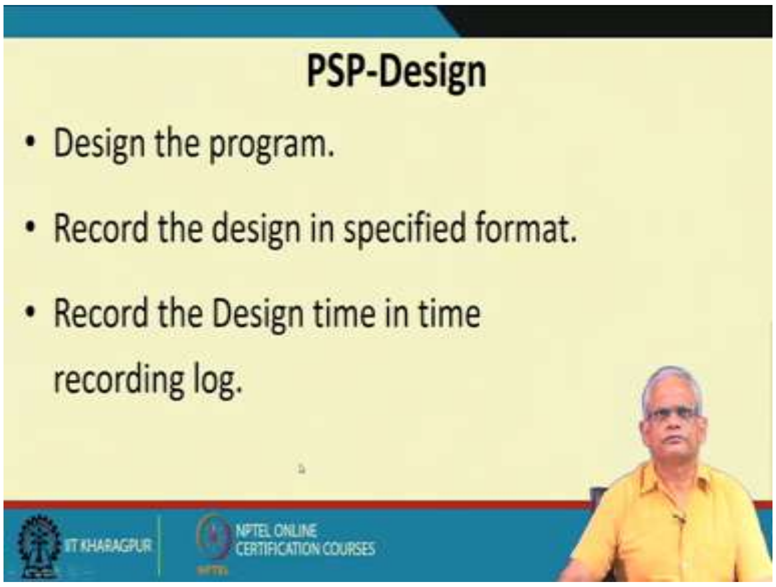
- Problem definition
- Estimate max, min, and total LOC
- Determine minutes/LOC
- Calculate max,min, and total development times
- Enter the plan data in project plan summary form
- Record the planned time in Log

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Other than the time management and keeping records or logs, the other aspect which is highlighted by the PSP is Planning. Starting with the problem the problem definition need to estimate what is the size of the work, what is the maximum minimum and estimated lines of code and from this must determine how much minutes it will take.

If a individual knows his coding speed that is how many minutes per lines of code, then he can determine the maximum minimum and total are the estimated development time. And these must be entered in a project plan summary form and once starts working must record the actual time the plan time and the actual time must be recorded.

(Refer Slide Time: 09:32)



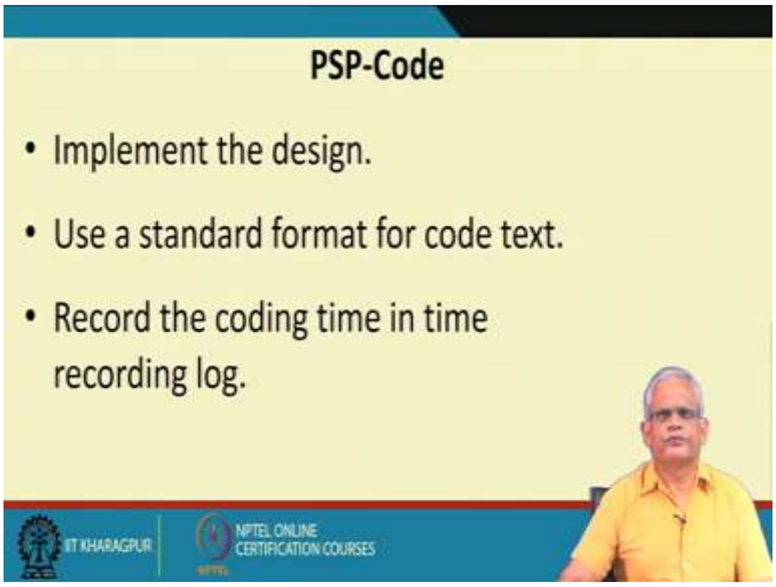
PSP-Design

- Design the program.
- Record the design in specified format.
- Record the Design time in time recording log.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

The design the specific format in which the design to be done. The developer must have his own standard which is of course final output must be compliant with the required standard of the team. But the way he goes about designing the specific format he must identify and then record the design time the time recording log.

(Refer Slide Time: 10:12)



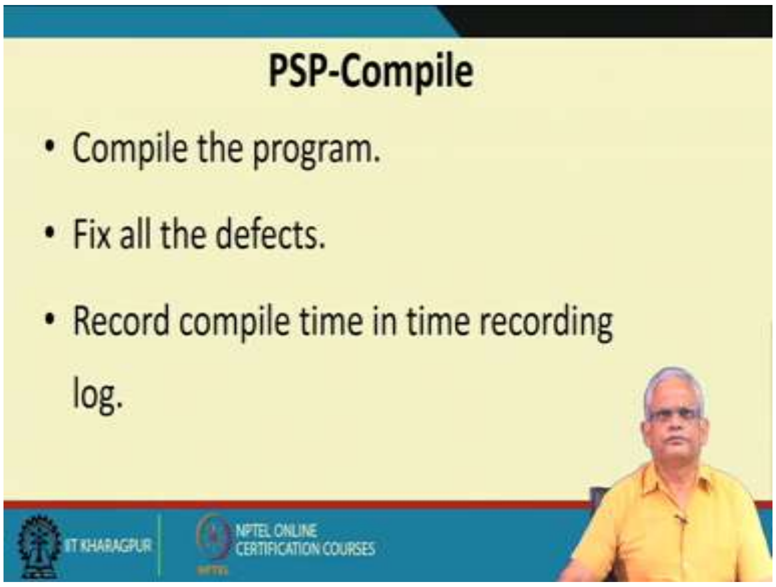
PSP-Code

- Implement the design.
- Use a standard format for code text.
- Record the coding time in time recording log.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

For the coding must implement the design and should have his own way to go about coding his own process some developers they first write the function prototypes and then identify the variables and define them and then write the code. Some other developers may feel comfortable in first writing a simple version of the function and slowly enhancing that adding more and more variables parameters and so on. But finally, the way the text is laid out that also must be for that must be standard and the coding time must be recorded in a time recording log.

(Refer Slide Time: 11:14)



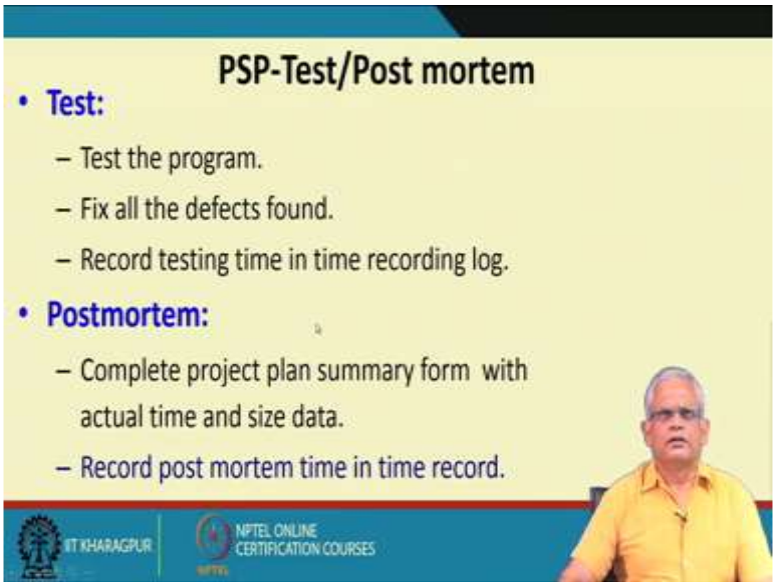
PSP-Compile

- Compile the program.
- Fix all the defects.
- Record compile time in time recording log.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

The Compile to remove all syntax errors, each time the program needs to be compiled and the syntax errors identified must be fixed until the program compile satisfactorily and this time is recorded in a log.

(Refer Slide Time: 11:34)



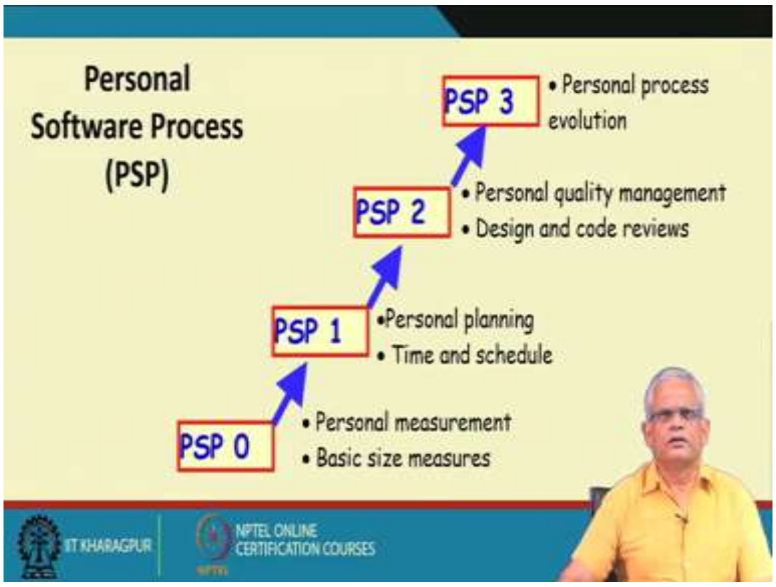
PSP-Test/Post mortem

- **Test:**
  - Test the program.
  - Fix all the defects found.
  - Record testing time in time recording log.
- **Postmortem:**
  - Complete project plan summary form with actual time and size data.
  - Record post mortem time in time record.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Then the testing here all the bugs have to be identified and fixed and the time is recorded in a log. And finally the postmortem where the project summary data that is the plan time and the estimated time are analyzed and then the observations are written down in postmortem.

(Refer Slide Time: 12:06)



Personal Software Process (PSP)

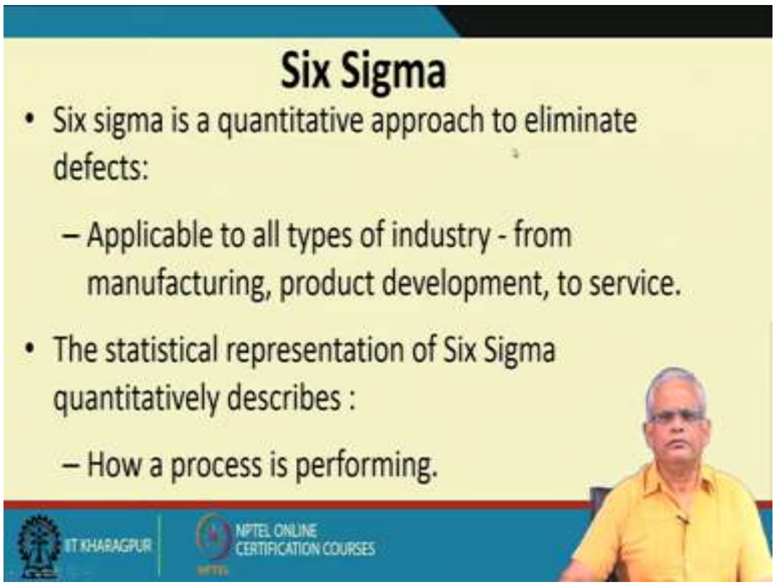
- PSP 0
  - Personal measurement
  - Basic size measures
- PSP 1
  - Personal planning
  - Time and schedule
- PSP 2
  - Personal quality management
  - Design and code reviews
- PSP 3
  - Personal process evolution

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

If we represent it schematically in the PSP there are four levels, PSP 0, PSP 1, 2 and 3 at the 0 level only the measurements are done, that is the personal measurement that is the how much time somebody takes for a line of code. It is time per lines of code the basic size measures that is given a work does he estimate that how much time will it take how many lines of code and so on.

In the PSP 1 level the basic planning is done, the personal planning that how to go about doing the work when to do which one this is planned and there is a schedule that by this time I must do this and so on. In the level 2 of PSP the personal quality management and the design and code review, the developer himself reviews his design and code and is written in the form of a log. And in the PSP 3 the personal process evolution and not only the personal process has been defined, but the opportunity to improve the process is identified and the process is continually changed at the level 3.

(Refer Slide Time: 14:06)



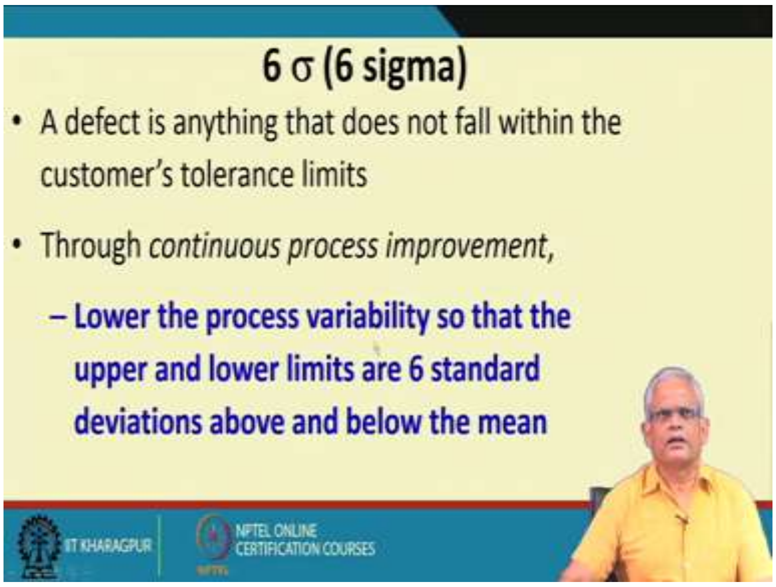
Six Sigma

- Six sigma is a quantitative approach to eliminate defects:
  - Applicable to all types of industry - from manufacturing, product development, to service.
- The statistical representation of Six Sigma quantitatively describes :
  - How a process is performing.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Now, let us look at very briefly at Six Sigma, the six sigma initiative is a quantitative approach to eliminate defects. Again this is applicable to all types of industry from manufacturing, product development, service. It is applicable to all industries and it incorporates statistical representation of the defects and based on that one can conclude that how a process is performing.

(Refer Slide Time: 14:50)



6 σ (6 sigma)

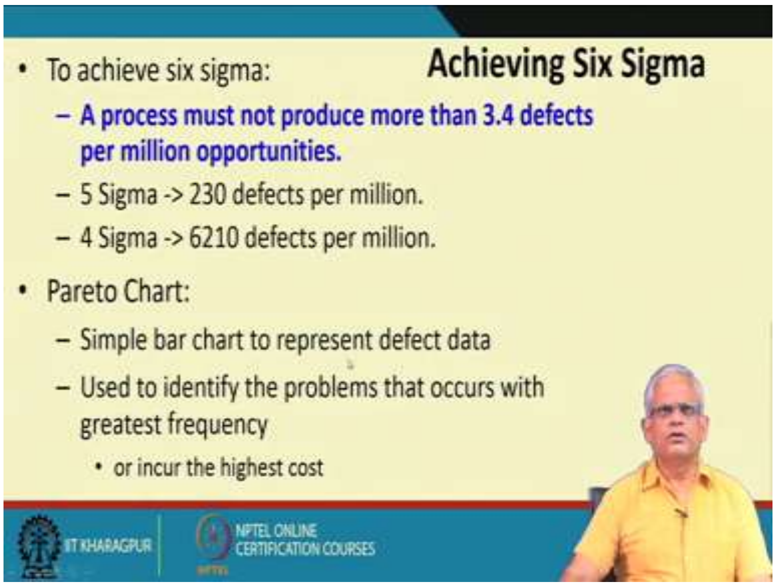
- A defect is anything that does not fall within the customer's tolerance limits
- Through *continuous process improvement*,
  - **Lower the process variability so that the upper and lower limits are 6 standard deviations above and below the mean**

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

A defect is defined in six sigma is anything that does not fall within customers tolerance limit. If something is a defect, but the customer is OK with that that is not really a defect.

In the six sigma it argues for continuous process improvement. So, that the variability in the process the upper and lower limits that are set this fall within six standard deviations above and below mean. We will not go into the detailed mathematical computations how to define the six standard deviations and so on.

(Refer Slide Time: 15:40)



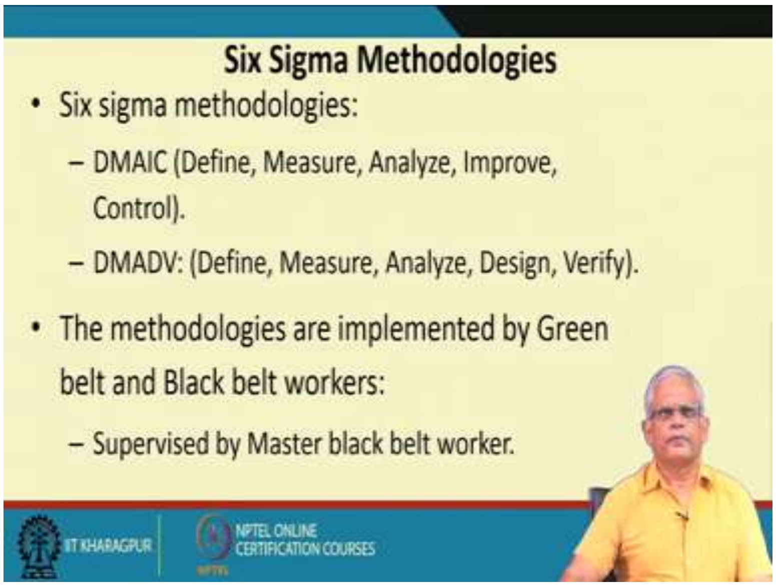
Achieving Six Sigma

- To achieve six sigma:
  - **A process must not produce more than 3.4 defects per million opportunities.**
  - 5 Sigma -> 230 defects per million.
  - 4 Sigma -> 6210 defects per million.
- Pareto Chart:
  - Simple bar chart to represent defect data
  - Used to identify the problems that occurs with greatest frequency
    - or incur the highest cost

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

But we will just say that to achieve six sigma a process must not produce more than 3.4 per million opportunities. If we look at definition of a 5 sigma or 4 sigma can see that the number of defects acceptable are much higher, a six sigma process must not produce 3.4 defects per million opportunities. Here the defects are represented using a Pareto chart and the defects that occur with the greatest frequency or that incur the highest cost are identified to take corrective action.

(Refer Slide Time: 16:32)



Six Sigma Methodologies

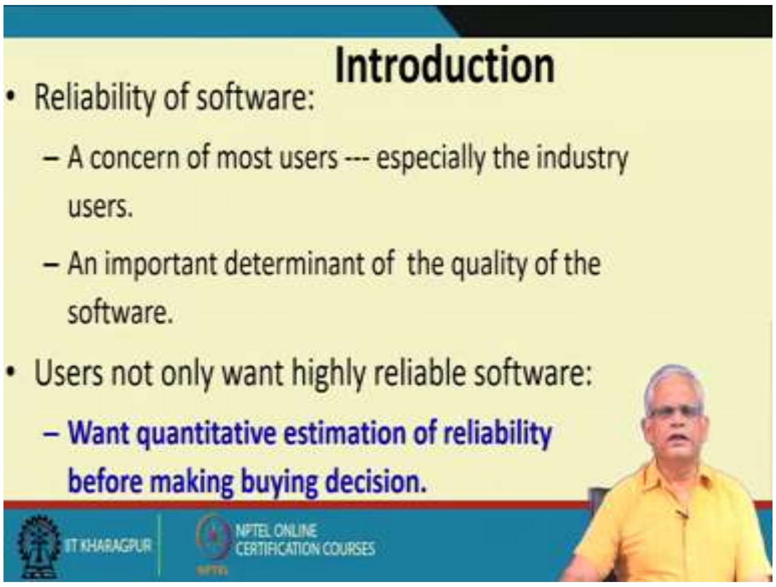
- Six sigma methodologies:
  - DMAIC (Define, Measure, Analyze, Improve, Control).
  - DMADV: (Define, Measure, Analyze, Design, Verify).
- The methodologies are implemented by Green belt and Black belt workers:
  - Supervised by Master black belt worker.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Several Six Sigma Methodologies are defined like define measure analyze improve and control DMAIC. Define measure analyze design and verify a DMADV and these are implemented by green belt and black belt workers supervised a master black belt worker. We will not go into the nitty gritty of six sigma methodologies we do not have that much time, in this course we just looked at a very overall idea of the six sigma.

We will now start looking at the software reliability issues. Now reliability is a very important issue in software, the customers they look for highly reliable software they would not buy software which is not reliable.

(Refer Slide Time: 17:31)



Introduction

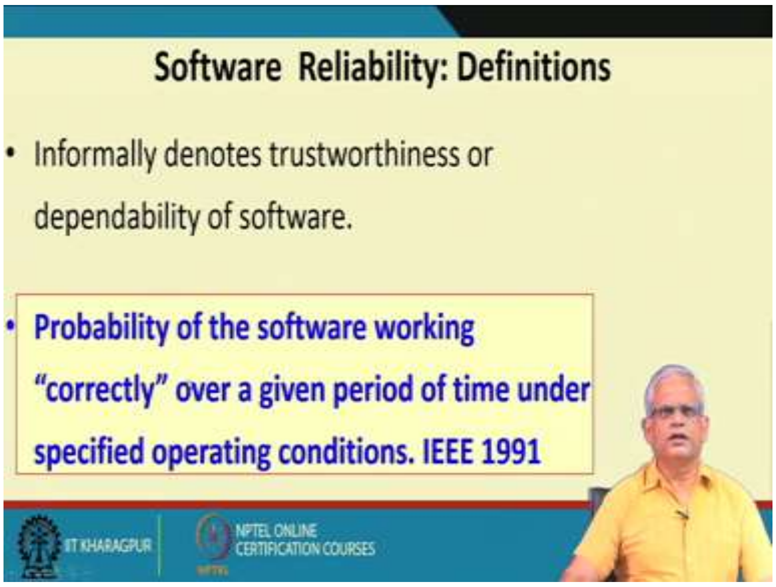
- Reliability of software:
  - A concern of most users --- especially the industry users.
  - An important determinant of the quality of the software.
- Users not only want highly reliable software:
  - **Want quantitative estimation of reliability before making buying decision.**

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

And therefore, it becomes a very important issue with the project manager to understand the nitty gritty of reliability and ensure that the final developed software has high reliability. Reliable software is a concern for all users, especially if it is used for some industry application it is very crucial and also reliability is a important determinant of the quality of the software. A software having poor quality sorry poor reliability cannot be said to be a quality software.

The users not only want highly reliable software, but they need a quantitative estimation to be given to them about the reliability before they can make any buying decision.

(Refer Slide Time: 18:50)



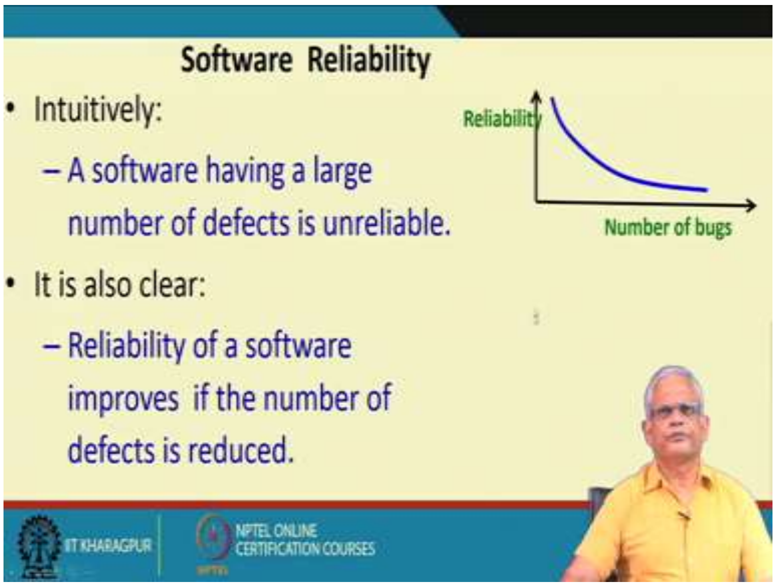
Software Reliability: Definitions

- Informally denotes trustworthiness or dependability of software.
- **Probability of the software working "correctly" over a given period of time under specified operating conditions. IEEE 1991**

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

But how do we define Software Reliability? Informally we can say that the reliability is basically the trust worthiness of the software or the dependability of the software. Intuitive definitions trust worthiness, dependability. But can we give a more formal definition of software reliability. IEEE document in 1991 defines reliability as the probability of software working correctly over a given period of time under specified operation conditions. So, here it is defined in terms of the probability of the software working correctly over a given period of time, under specific operating conditions.

(Refer Slide Time: 19:54)



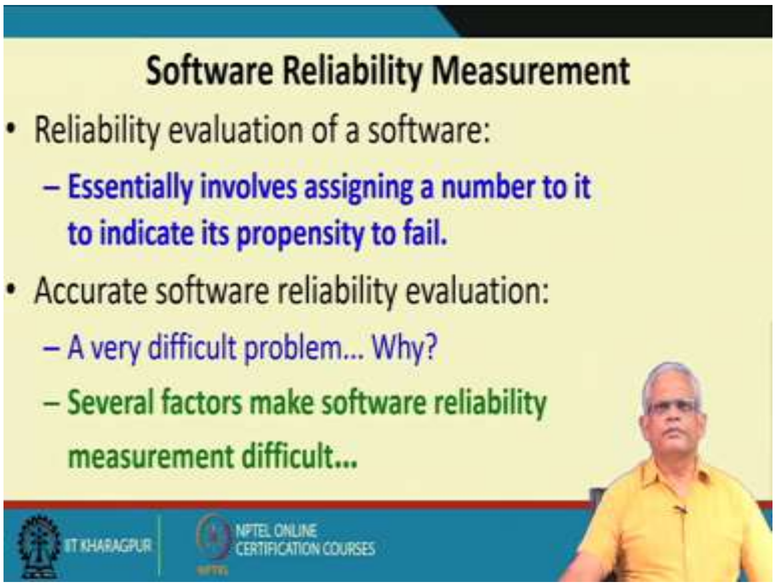
Software Reliability

- Intuitively:
  - A software having a large number of defects is unreliable.
- It is also clear:
  - Reliability of a software improves if the number of defects is reduced.

Reliability
Number of bugs
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

To get some intuitive idea about software reliability, a software that has large number of defects is clearly unreliable. And also we know that if we remove some bugs that is as the bugs are identified and fixed, the reliability improves. When there are large number of bugs the reliability is low. But as the number of bugs reduces the reliability improves.

(Refer Slide Time: 20:34)



Software Reliability Measurement

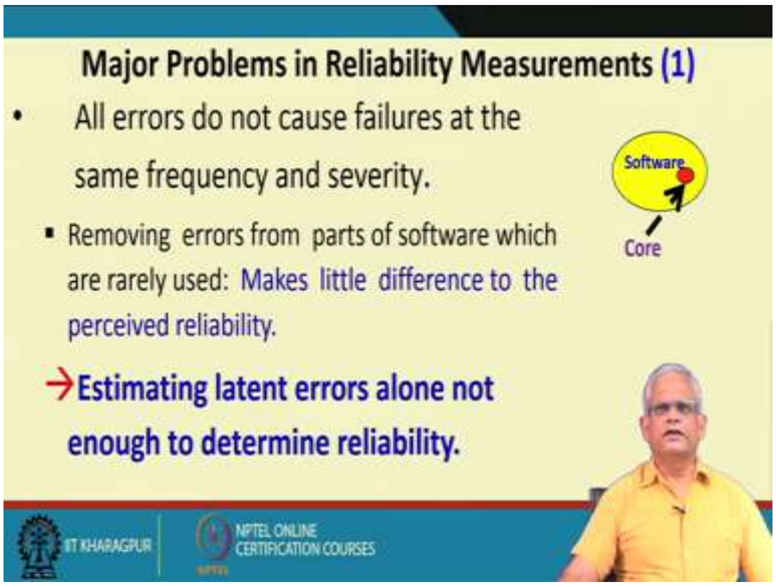
- Reliability evaluation of a software:
  - Essentially involves assigning a number to it to indicate its propensity to fail.
- Accurate software reliability evaluation:
  - A very difficult problem... Why?
  - Several factors make software reliability measurement difficult...

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

For Reliability Measurement we need to give a number to the software, that will identify or indicate it is likelihood of failing. For hardware it is a relatively simple issue how to compute the reliability of hardware you keep on operating the hardware, make observations and the failure and that gives the reliability of the hardware. But for software it is a difficult problem, for hardware just put it on use for a long period of time and then just give the reliability measurement of the hardware.

But for software it is difficult let us first understand what is causing or making the software reliability evaluation a much more harder problem than software. There are actually several factors several important factors. We will look at four five of them which make software reliability measurement much harder than hardware reliability measurement.

(Refer Slide Time: 22:07)



Major Problems in Reliability Measurements (1)

- All errors do not cause failures at the same frequency and severity.
  - Removing errors from parts of software which are rarely used: Makes little difference to the perceived reliability.

→Estimating latent errors alone not enough to determine reliability.

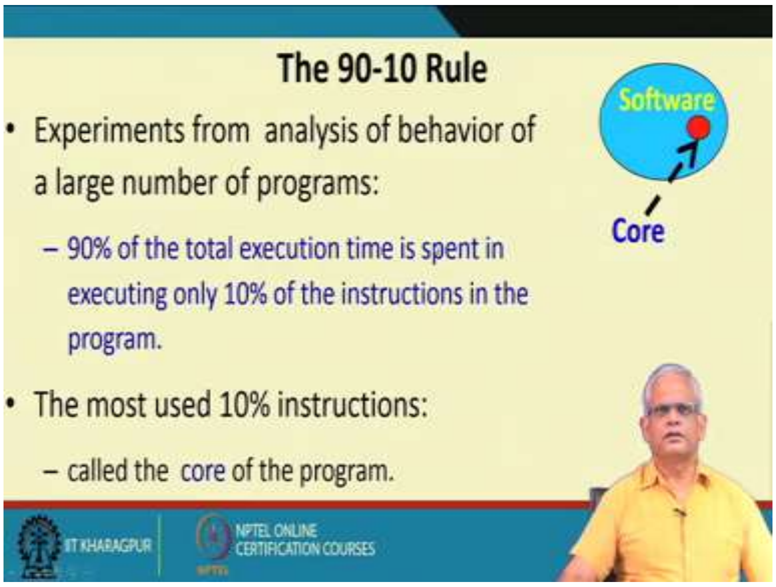
Software
Core
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

One issue is that if there are hundred bugs in a program, all bugs do not cause failures in the same frequency and same severity. Some of the bugs they keep on making the software fail every now and then. Whereas, some other bugs they rarely make the system fail. We can think of a software as it operates there are some areas of the program code which get executed again and again, very frequently some part of the code gets executed.

And therefore, if that part of the code has a bug this will normally show very frequently, but if some other part of the code which is rarely used has a bug this will not appear so frequently. If this is the software some part of the code gets executed very frequently this is called as the core of the program. For every program we can identify the core, there are tools available called as profilers which can identify the core and therefore if bugs are present here in the core they will cause it to fail very frequently.

So, if there are ten bugs present in the core or let us say just one bug present in the core, the software may exhibit much higher failure rate than another software which has let us say hundred bugs in the noncore part. And for this reason just estimating the number of errors that are remaining in the software is not good enough to predict how frequently it will fail. We must also identify whether it is in the core part or in the noncore part.

(Refer Slide Time: 24:39)



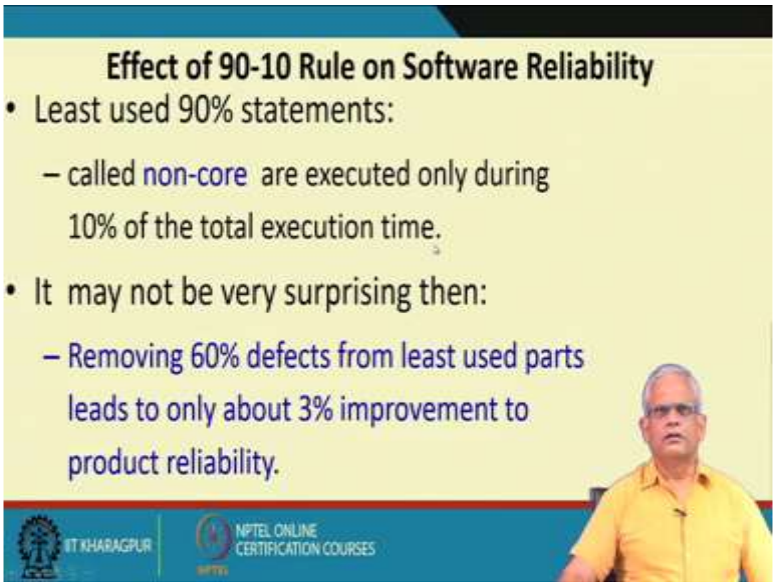
The 90-10 Rule

- Experiments from analysis of behavior of a large number of programs:
  - 90% of the total execution time is spent in executing only 10% of the instructions in the program.
- The most used 10% instructions:
  - called the core of the program.

Software
Core
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

The 90-10 rule says that 90 percent of the execution time of a program is spent in executing 10 percent of the code. For any program this can easily be identified by a profiler tool for Unix the tool name is p r o f prof and we can easily identify the core of a program. And if the bugs are in the core of the program the reliability will be very low it will fail very frequently. But even if much larger number of bugs are present in the noncore part of the program the program may not fail so frequently.

(Refer Slide Time: 25:36)



Effect of 90-10 Rule on Software Reliability

- Least used 90% statements:
  - called non-core are executed only during 10% of the total execution time.
- It may not be very surprising then:
  - Removing 60% defects from least used parts leads to only about 3% improvement to product reliability.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Study shows that removing 60 percent defects from the least used parts leads to only about 3 percent improvement in the reliability. But if these are used from the if they are removed from the most used parts or the core of the program then the reliability will improve by a significant factor. Just imagine that if there are hundred bugs in the least used parts that is noncore and you remove 60 of them the reliability improves by only 3 percent.

So, to estimate the reliability of a software just knowing the number of bugs is not good enough there are several other factors which make the reliability evaluation much more complicated. We look at these issues in the next lecture, for this lecture we are almost at the end of the lecture hour, we will continue from this point in the next lecture.

Thank you.