Personal Software Process(PSP)

Personal Software Process

Analogical Reasoning

World = Software Development Project

Birth = Me joining the development team

The Personal Software Process (PSP)

- The **software process** is about making software engineering groups/ teams work to the best of their abilities.

- The **personal software process** is about making individual engineers work to the best of their abilities.

- Through analysis of practical application of the process, the process should be changed for the better.

The Personal Software Process (PSP)

- Software engineers should accept responsibility for the quality of their work.
- Software engineers can do this only if they have a way of evaluating quality and improving quality (through experience)
- The software process improves individual engineers to some extent.

Objective of PSP

- Improve their estimating and planning skills.
- Make commitments and schedules they can keep and meet.
- Manage the quality of their projects.
- Reduce the number of defects in their work.

Principles of Personal Software Process

- Every engineer is different. For software engineers to become more active, they should plan their work and base these plans on their personal data.
- To improve their performance, software engineers should personally use regular and well-defined processes.
- For software engineers to produce quality software products, they should feel personally responsible for the quality of the products they are making. Superior software products are never created by mistake but by striving to do quality work.

Principles of Personal Software Process

- It’s cheaper to trace and fix defects earlier than later.
- It’s easier to prevent errors than finding and fixing them.
- The cheapest and fastest way to do any task is doing it in the right direction.

Criticism of the PSP

The PSP is not universally accepted:

- some think it is a good idea in theory but not in practice
- some think that it is not flexible enough
- some think that it is too time consuming
- some think it should be up to individuals to find their own way of working

- The PSP fails, when it is used by the team to criticize individuals.

PSP Process Evolution

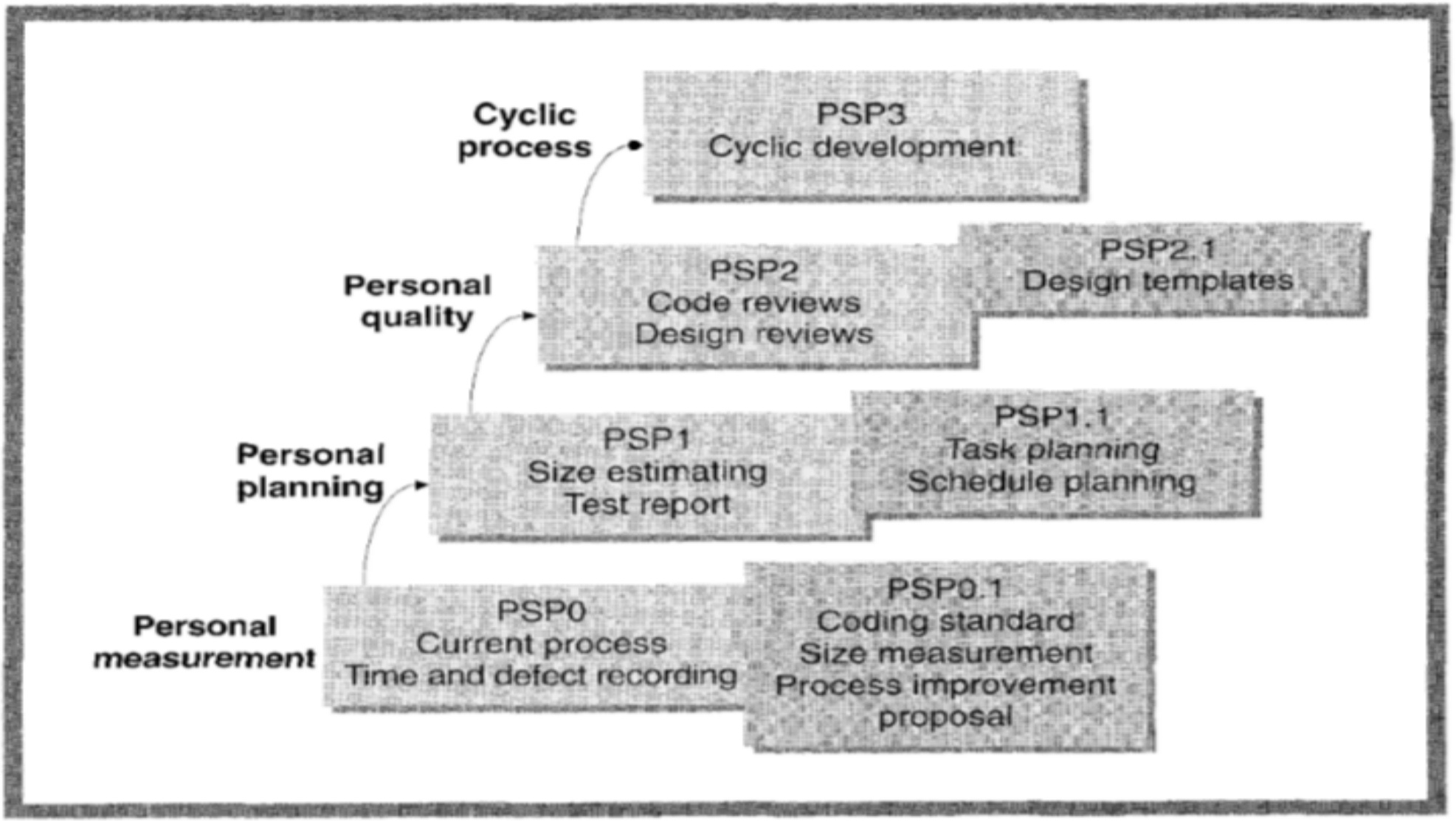


Figure 1. PSP process evolution.

PSP0: Personal Measurement

- Engineers gather data on the time they spend by phase and the defects they find.
- Generates real, personal data and provides the base benchmark for measuring progress
- PSP0 adds a coding standard, size measurement and a process improvement proposal

PSP Basic Measures

- **Development Time**: measured in minute using a *time recording log* designed to account for interruptions
- **Defects**: any change to the design or code to get the program to compile or test correctly; recorded in a *defect recording log*
- **Size**: lines of code, used primarily for estimating development time; new, modified and re-used code is distinguished.

Line of Code (LOC)

- Source lines of code (SLOC), also known as lines of code (LOC), is a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code.

- **Declarations, Actual code including logic and computation.**

- **Blank Lines, Comments**

Line of Code (LOC) Example

LOC = 4

Python

```
1) #addition of two number
2) def add(a,b):
       3) return a+b;
4) #Function calling
5)
6) res=add(2,3)
7) print(res)
```

LOC= 12

C Language

```
1) #include<stdio.h>
2) int add (int,int);
3) void main()
4) {
5) int res;
6) res=add(2,3);
7) printf(res);
8) }
9) \\addition of two number
10) int add (int a,int b)
11) {
12)
13) return a+b;
14) }
```

PSP1: Personal Planning

- This step must introduce some method for estimating sizes and development times for new programs based on personal data
- The methods employed are usually (should be) based on **linear regression** with prediction intervals to indicate size and estimate quality.
- *PSP1.1 adds schedule and task planning*

Size Measurement

- The planning process starts with an estimate of the job size.
- By estimating the size of the product that you plan to build, you can better judge the amount of work needed to build it.
- Before you can estimate the product size, you need to select an appropriate size measure.

Size Measurement Criteria

The criteria for a size measure are that it must be

- useful for planning
- directly countable/measurable

Useful for Planning

- A primary goal in planning is to estimate the effort for a job.
- For a size measure to be useful during planning, it must help you to estimate the effort for the job.
- To help you estimate the effort for the job, there must be a relationship between the size measure and the effort for the job.
- If a relationship exists between a size measure and effort, then the size measure can be used for planning.
- The next slide illustrates such a relationship for writing pages of text for a book.

Directly Countable

- The size measure must be directly countable or measurable in the product.
- Examples of directly countable measures are
  - pages
  - paragraphs
  - diagrams
  - graphs
- Examples of something that is not directly countable in a document are themes or topics.

Effort Measure

- While the size measure depends on the job, the effort measure is always time.
- The concerns with measuring time are these.
  - What is included in the time measure?
  - What precision is used to measure time?

What is Included in the Time Measure?

- Since you are looking for a relationship between size and effort, only the time spent actually working on a task (“time on task”) should be included.
- Interruptions such as phone calls, rest breaks, and office conversations
  - are highly variable
  - are not related to the job
  - must be excluded from task time in order to determine the size versus time relationship
- Most engineers find that they spend less than 50% of their time on task.

Time Measurement Precision

- Effort on projects is often measured in person months.
- While this is useful for financial accounting systems, it is not useful for tracking, analyzing, or improving personal processes.
- People rarely work on one thing for more than an hour or two.
- In PSP, effort is measured in minutes.
  - This measure is precise enough.
  - Once you decide to precisely measure time, the units do not make much difference.

Selecting a Size Measure -1

- Start with some historical data from past projects.
  - development effort
  - product characteristics (candidate size measures)
- Rank the products by development effort.
- See what characteristics distinguish those products that took the greatest effort from those that took the least.

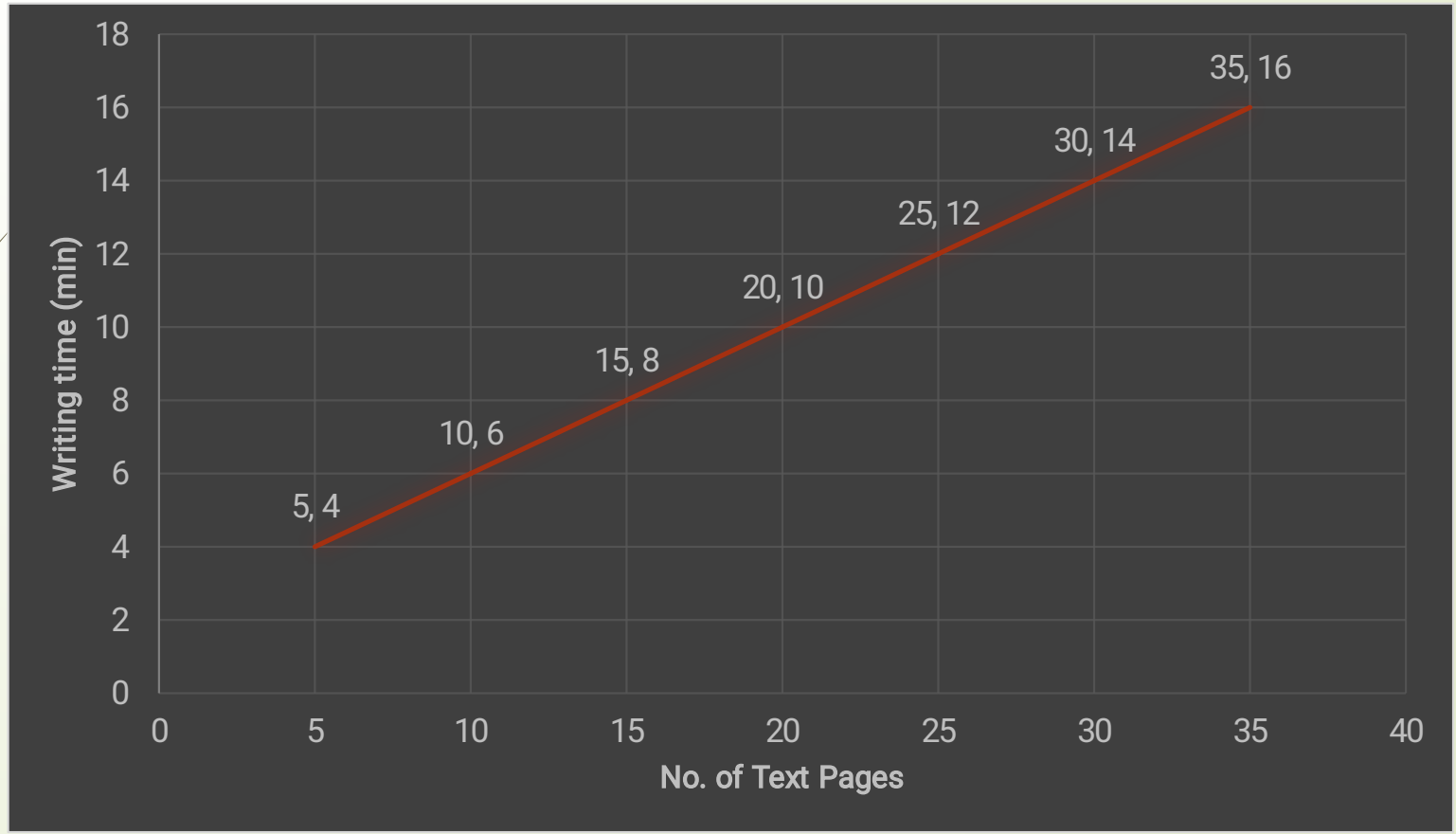
Selecting a Size Measure -2

- Determine if the candidate size measure is related to effort.
  - Create a graph with the candidate size measure on the horizontal axis and effort on the vertical axis.
  - Plot each pair of historical data (candidate size measure, effort) as a point on the graph.
  - If the points define a line, then the candidate size measure is related to effort and can be used for planning.
- There may be no single best size measure. A combination of size measures could be needed.

Ex: Text Pages versus Writing Time

No. of Text Pages	Writing Time (min)
5	15
10	30
15	45
20	60
25	75
30	90
35	105

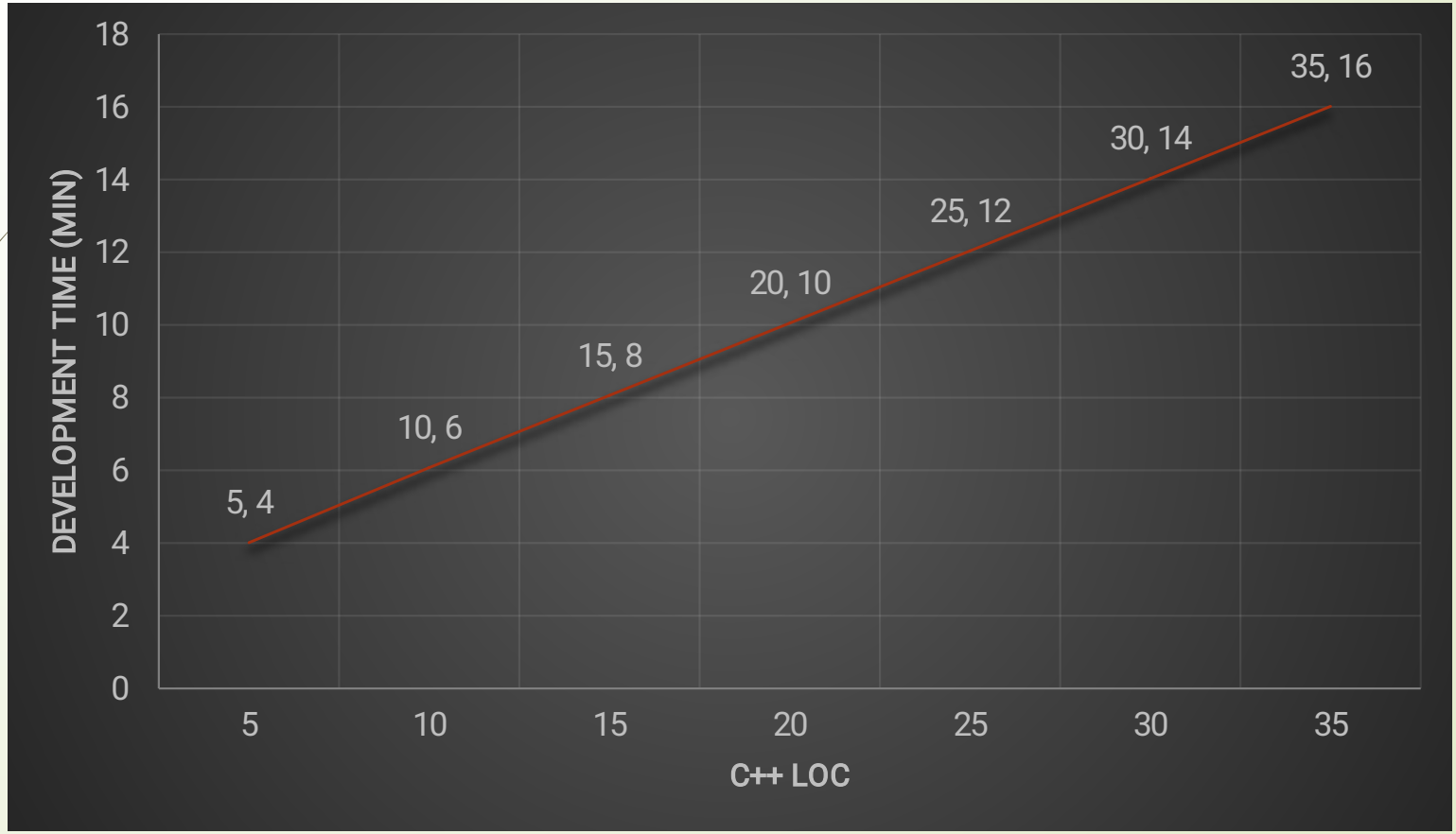
Ex: Text Pages versus Writing Time



Ex: LOC versus Development Time

LOC	Development Time (min)
5	4
10	6
15	8
20	10
25	12
30	14
35	16

Ex: LOC versus Development Time



Estimation of development time using regression model.

$$y = a + bx$$

$$a = \frac{\sum Y \cdot \sum X^2 - \sum X \cdot \sum XY}{n \cdot \sum X^2 - (\sum X)^2}$$

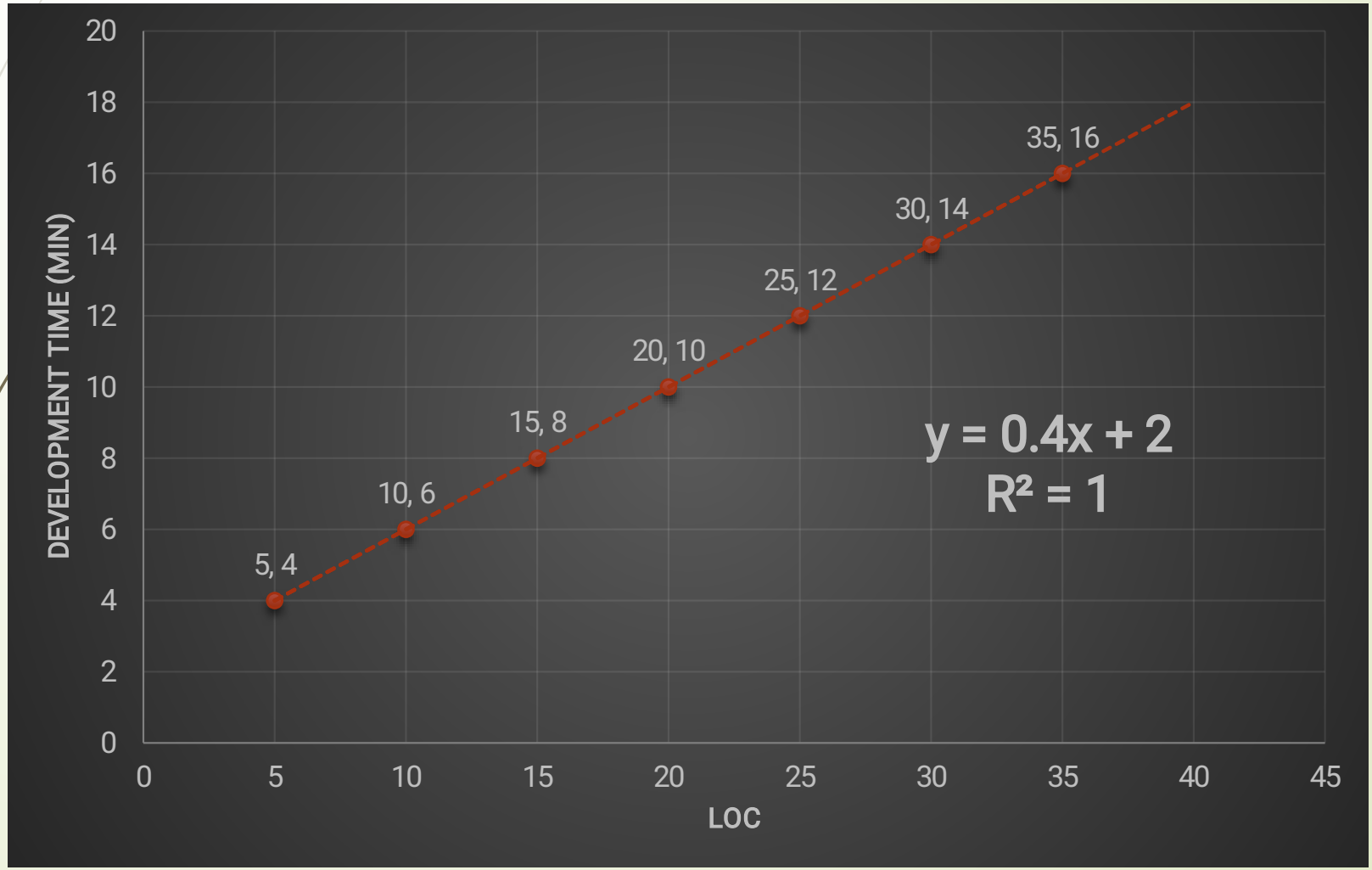
a= 2

b= 0.4

$$b = \frac{n \cdot \sum XY - \sum X \cdot \sum Y}{n \cdot \sum X^2 - (\sum X)^2}$$

y= 2+0.4x

Ex: LOC versus Development Time



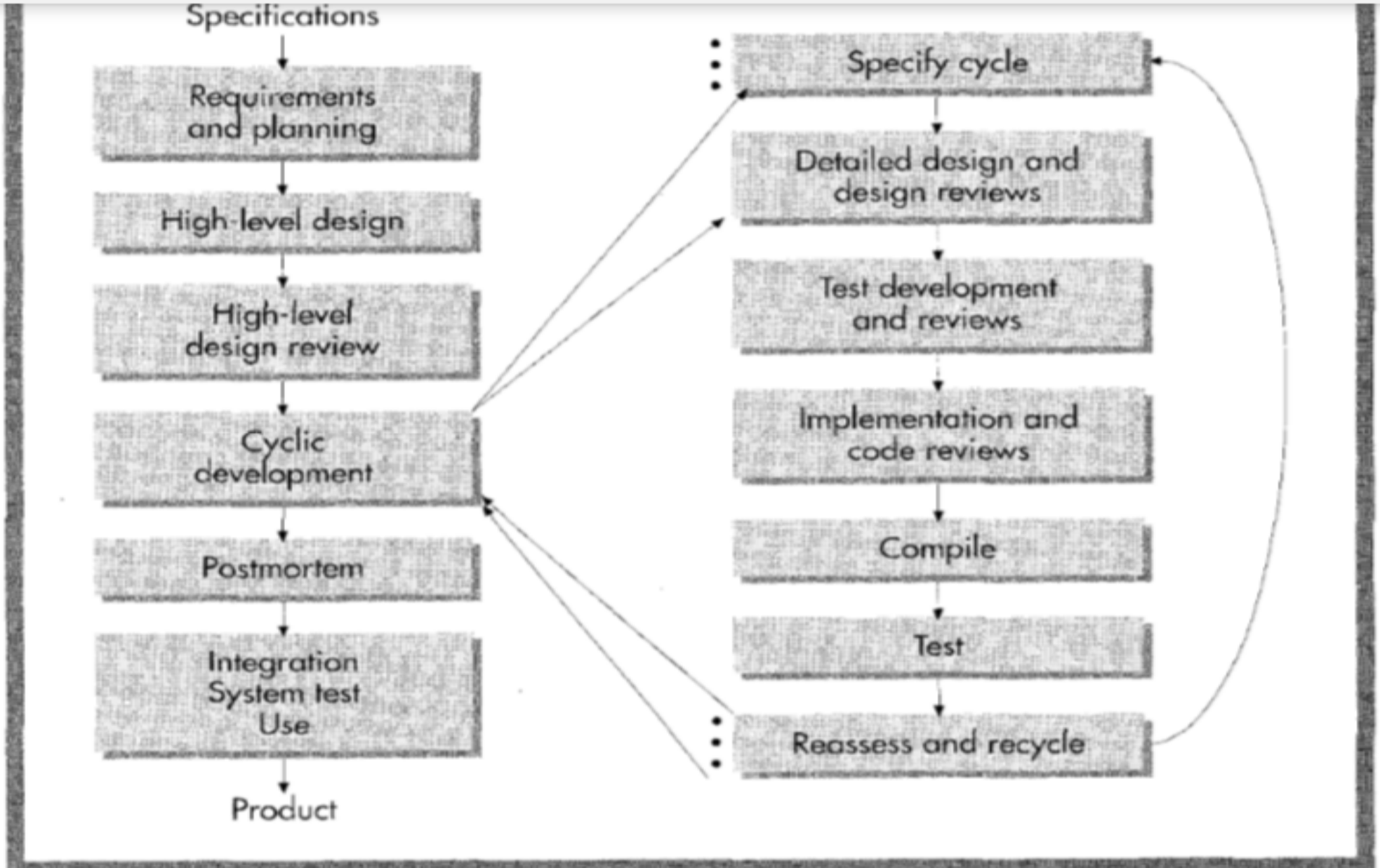
Estimating Effort Directly

- If you cannot find a good size measure, you may have to estimate development time directly.
  - Collect a lot of historical data.
  - Compare the new work with past work.
  - Judge the likely development time.

PSP2: Personal Quality

- This step introduces defect management
- Using data from PSP exercises, engineers construct and use checklists for design and code review
- From their own data, they see how checklists help personal reviews
- *PSP2.1 adds design specification and analysis techniques along with defect prevention, process analyses and process benchmarks*

PSP 3: Cyclic Process



PSP Quality Strategy

- Defects are basic quality measure
- Engineers should:
  - •remove them
  - •determine their cause (type)
  - •learn to prevent them
- PSP uses private review with the goal of finding all defects before 1st compile and test