Experiences in Implementing Defect Prevention activities in Software Product Development Life Cycle

Tathagat Varma
Philips Software Centre
Bangalore, India

PHILIPS
SOFTWARE CENTRE

PHILIPS
Let's make things better.

Why do defects get introduced in our products ?

- Systemic problems
  - Incomplete / incorrect processes
  - Lack of education / training
  - Lack of communication
  - Transcription Error
- Misexecution - natural variations in personal productivity and not subject to software process improvement activities. Various reasons include time pressure, “bad days”, etc.

What is the cost to correct defects ?

Industry data suggests a 10-fold increase (in the order of magnitude) to correct defects in phase next to the one where it was introduced.

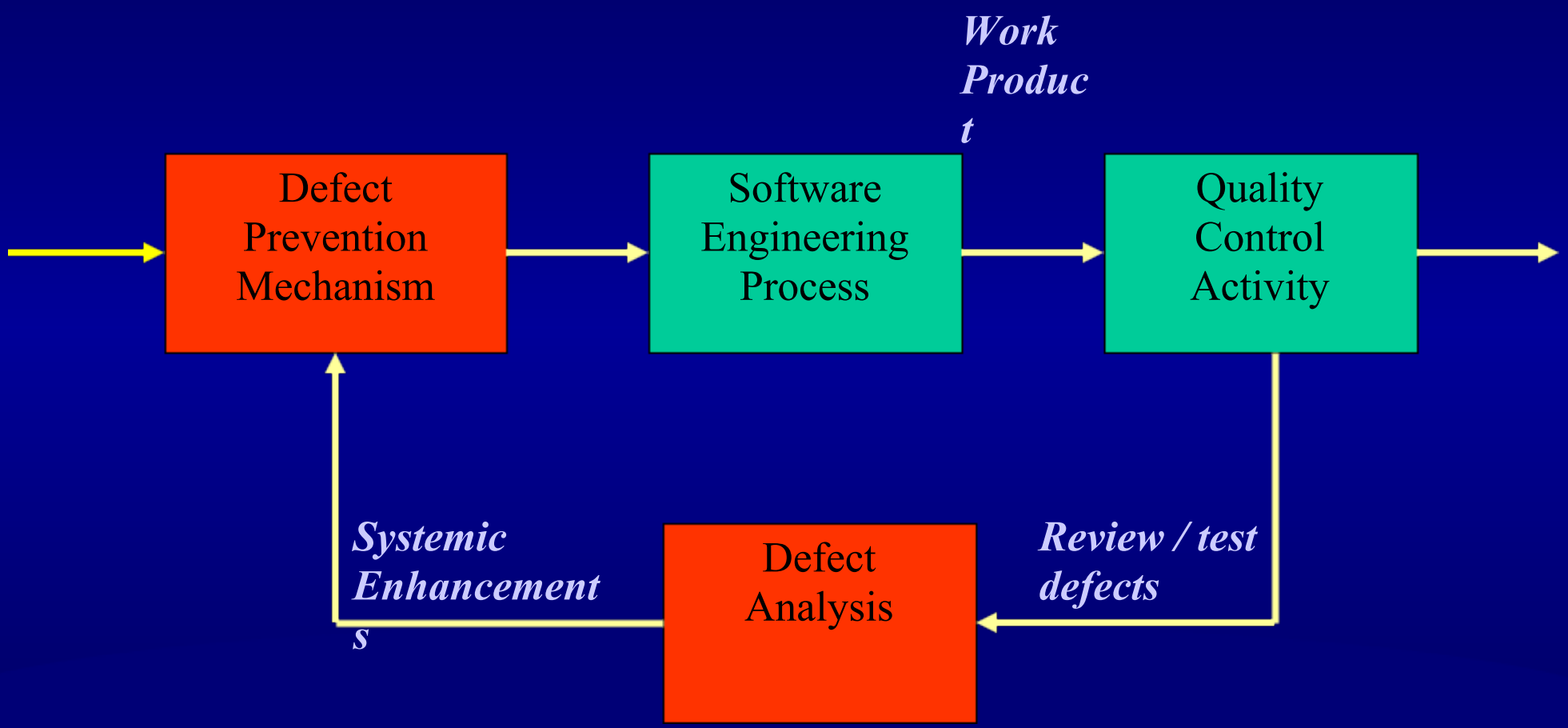
For example, for every defect introduced in the Requirements phase but discovered in the Design phase, the cost to correct goes up 10-fold !

How to prevent defects ?

- Corrective action is reactive - the cost of wastage has already been incurred.
- Preventive action is proactive - reuse the lessons learnt, in the process of making mistakes, to avoid making same mistakes again

- Simple principle of closed-loop management: analyze defects - establish root cause - make systemic enhancements. Keep refining the process till 80/20 gains are visible.

What is Defect Prevention Model ?



What are various Defect Prevention Mechanisms ?

- Training
- Communication
- Well-defined processes - make it difficult to introduce defects
  - checklists for entry criteria / exit criteria
  - clear-cut progression criteria from one phase to another
- Automation of routine manual operations

What is Defect Analysis ?

Objective: 80/20 Return-on-Investment

- Find those 20% defects that are (potentially) responsible for 80%

of the "cost of chronic wastage"

- Perform a defect analysis based on your needs - ODC, Phase-wise, module-wise, etc
- Identify systemic causes behind these defects
- Install systemic enhancements (or "process improvements") in software processes used

Other Defect Prevention Techniques

- Preview Meetings
- Post Mortem Meetings
- Checklists
- Prototyping
- Concurrent Coding and Unit Testing
- Reuse
- Automation

Experiences from Project X

- Q1’97: Taken-over from Sweden
- Q2 ‘97: Unstable product; more testing leading to more defects; one month of testing yields over 1000 defects
- Q3’97: Reverse engineered the software; Defined software processes
- Q4‘97: Initiated defect analysis
- Q1‘98: Defined Defect Prevention Process
- Q2’98: Prototyped the process
- Q3’98: Delivered software with zero-known defects

Lessons learnt

- Defect Prevention can be initiated anytime in any project
- Defect Prevention has to be continuously stoked - it is a dynamic process
- There is nothing like “one-size-fits-all” in defect prevention mechanism
- Mature approach for avoiding personal conflicts when defects are discovered
- Expected time to start getting improvements is around 6 months
- Ownership by Project Leader is a critical success factor

PHILIPS SOFTWARE CENTRE

PHILIPS

Let's make things better.

Problems faced

- Culture shift - takes times, but 6 months are enough to start seeing improvements
- Non-repetitive projects / tasks do not give enough opportunities for systemic improvements, but certain activities can still be performed.
- Tight delivery schedules that were tempting us not to do these activities

Conclusions

- Measure your cost of wastage - it is eating into your IFO !
- It is possible to prevent making defects even without a formal SPI program addressing higher maturity levels
- It is possible to measure the improvements against your investments so that you can strike a balance
- Defect Prevention principles can be applied to any type of project, even non-software projects.
- No matter what stage of development / maintenance a project is

in, Defect Prevention can always be initiated.

PHILIPS SOFTWARE CENTRE

PHILIPS

Let's make things better.

Any questions ?

PHILIPS
SOFTWARE CENTRE

PHILIPS
Let's make things better.