

*Test Metrics*  
door Rob Baarda, Sogeti

---

17 september 2009

voordracht georganiseerd door het



TECHNOLOGISCH INSTITUUT  
*Discussiegroep Software Testing*

met de steun van



systematically delivering success

quasus



---

Ingenieurshuis - K VIV, Antwerpen

# Test Metrics

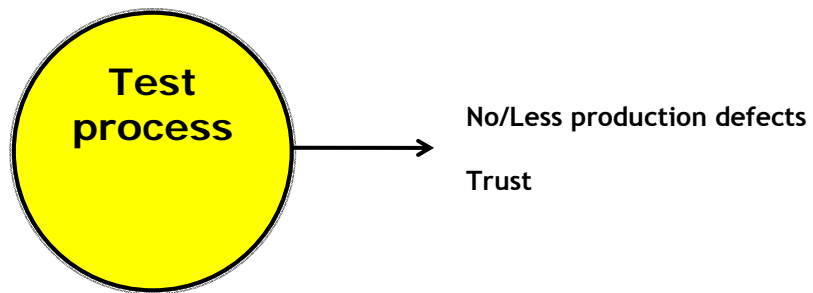
Rob.Baarda@Sogeti.nl



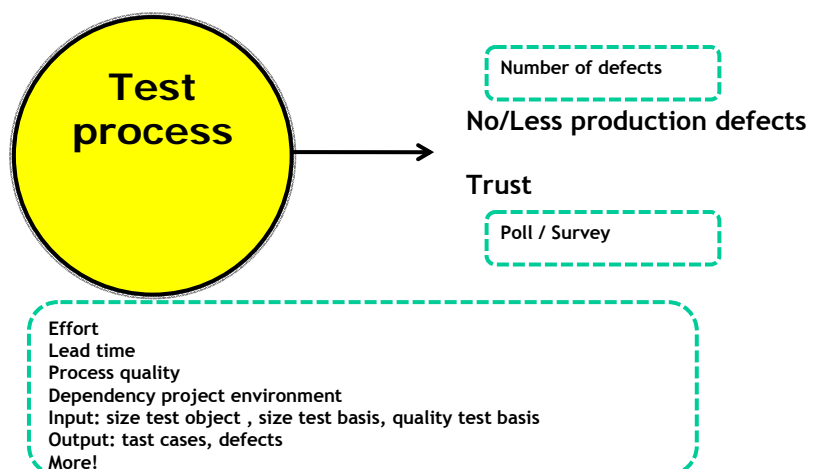
## Agenda

- **Introduction**
- **Theory**
- **Experience: metrics in practice**
- **How to use, pitfalls**
- **More practical data**
- **Wrap up**
- **Bonus: Value of testing**

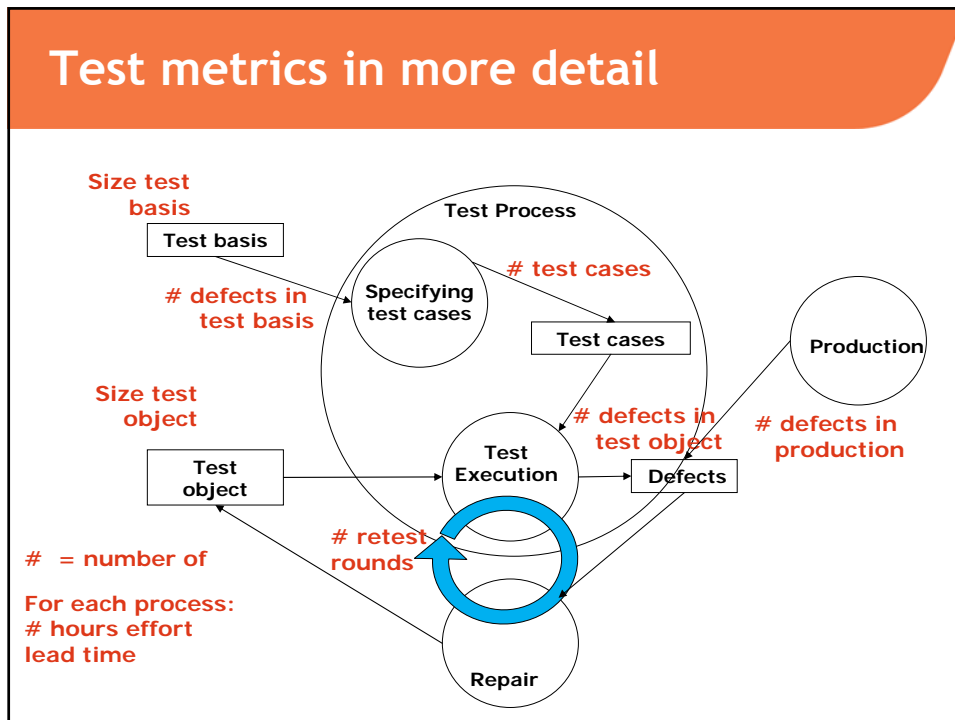
## Test =



## Test metrics =



## Test metrics in more detail




## Deductible metrics

- Effort estimation = # hours /size (FP, KLOC)
- Productivity = # test cases / # hours
- Efficiency =  
# defects / (# hours or # test cases)
  - > Specification
  - > Test execution
  - > Retest of repaired defects
- DDP Defect Detection Percentage (Europe)
- DRE Defect Removal Efficiency (USA)
- Defect injection rate for rework
- Damage prevented in €?

WHAT to know?  
HOW to get data?  
HOW to organize?

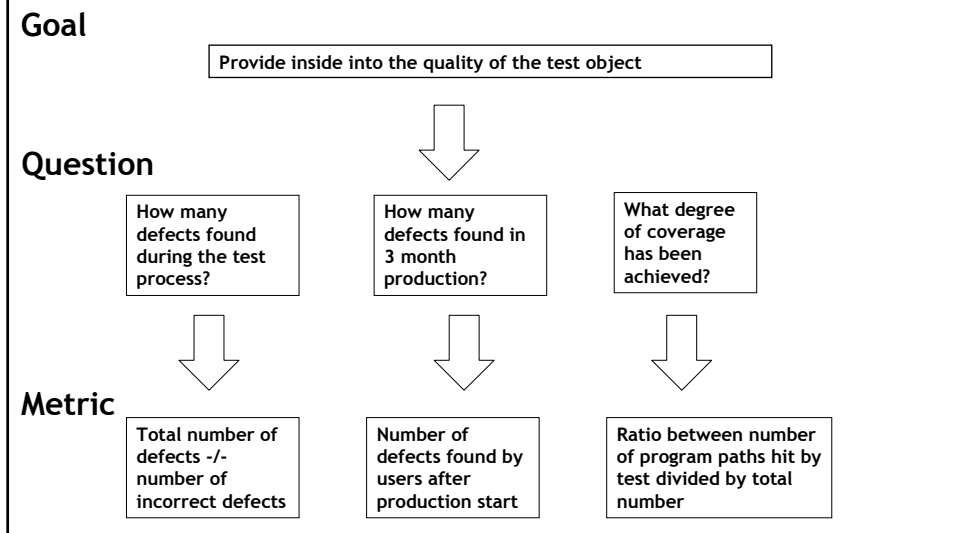
## Agenda

- Introduction
- Theory 
- Experience: metrics in practice
- How to use, pitfalls
- More practical data
- Wrap up
- Bonus: Value of testing

## What to know?

- Gathering all that data costs...
- Who wants what?
- Use the Goal – Question – Metric approach  
Basili, Galdierra, Rombach 1994

## GQM as example



## Agenda

- Introduction
- Theory
- Experience: metrics in practice ←
- How to use, pitfalls
- More practical data
- Wrap up
- Bonus: Value of testing

## Experience: Dutch test metrics group

- Dutch initiative to gather test metrics
- Parties involved



**NESMA**

Netherlands Software Metrics Association



**Testnet**

Dutch Testing community



**LaQuSO**

Laboratory for Software Quality

Universities Eindhoven & Nijmegen

Approach

Goal Question Metrics (GQM)

## Goals

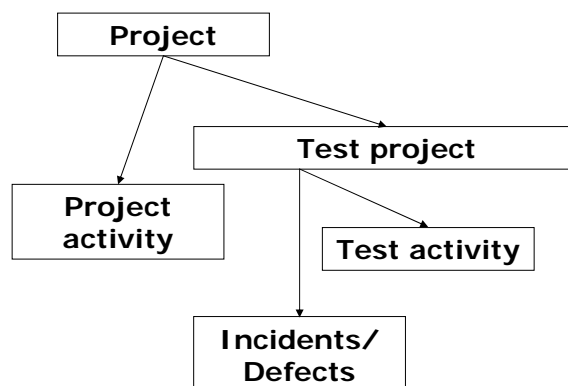
1. Test manager  
Support in planning and controlling the testing project
2. Organization Benchmark around
  - > Test process
  - > Test products
  - > IT-productsTo improve test process, IT process

## Some Questions

- **Test manager**
  - > Number of test cases needed for my project?
  - > What percentage of the project team should be allocated to testing?
  - > How many retests are executed?
- **Organization Benchmark**
  - > What is the defect detection & removal efficiency (at what phase)?
  - > What test coverage do I need to ensure adequate testing?
  - > How many defects does development insert when repairing other defects?

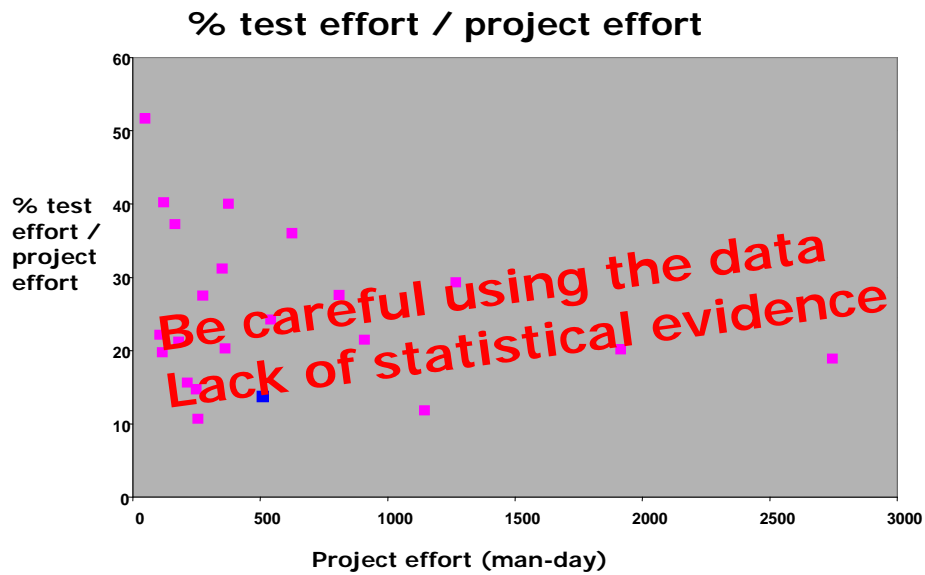
13

## Structure

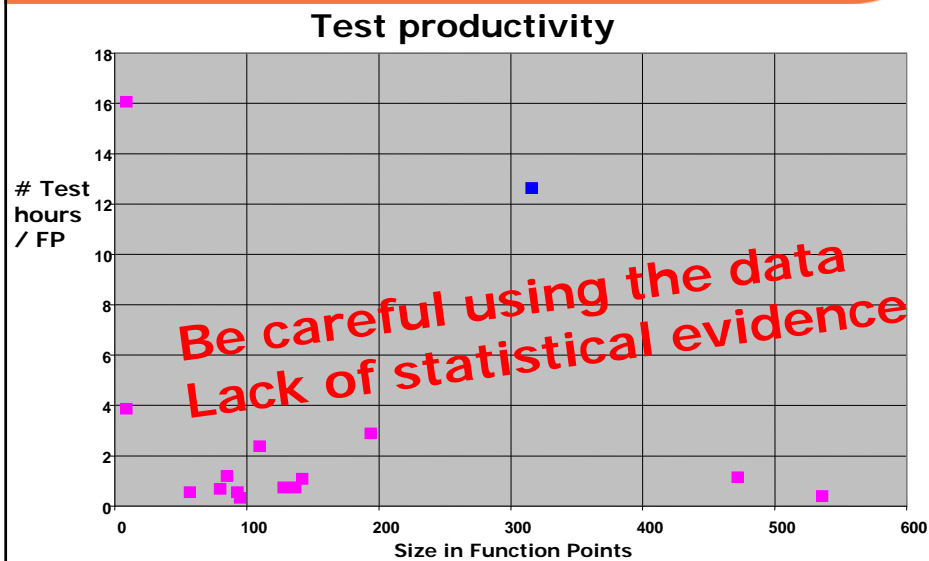


Be careful using the data  
Lack of statistical evidence

## Feedback example test effort

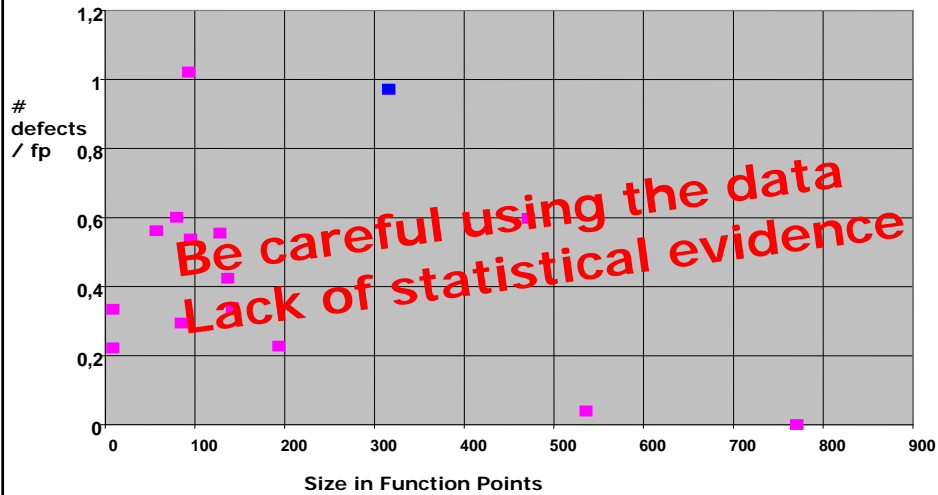


## Feedback example test productivity



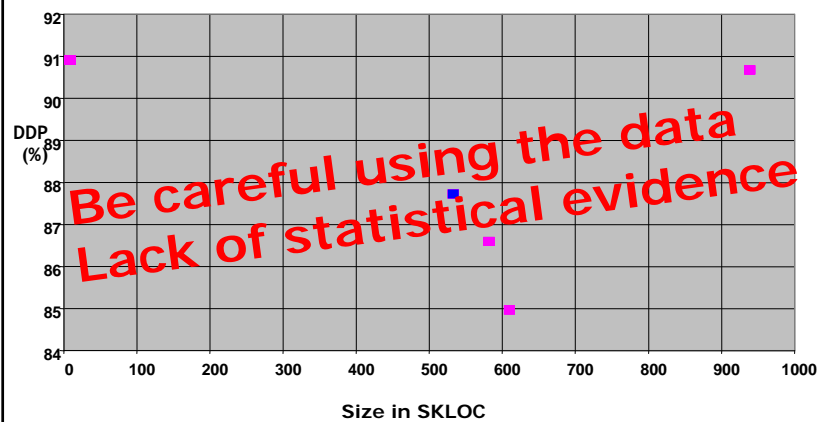
## Feedback example defects per fp

Number of defects / function point



## Defect Detection Percentage Defect Removal Efficiency

Defect Detection Percentage  
Defect Removal Efficiency



## More goals in practise

- **Predict planning**
- **Benchmark**
- **Measure the value of testing**
  - > **Number of defects in production**
  - > **Prevented damage**
- **Show gain in test productivity by eg. Outsourcing**
- **Prove improvement of test process**
- **Measure personal productivity**

## Metrics in general

- **ISBSG**
  - > **Launch of test metrics September 2009**
- **Commercial**
  - > **SEER, Dan Galorath**
  - > **SPR, Capers Jones**
  - > **QSM, Putnam family**
  - > ...

## Agenda

- Introduction
- Theory
- Experience: metrics in practice
- How to use, pitfalls
- More practical data
- Wrap up
- Bonus: Value of testing



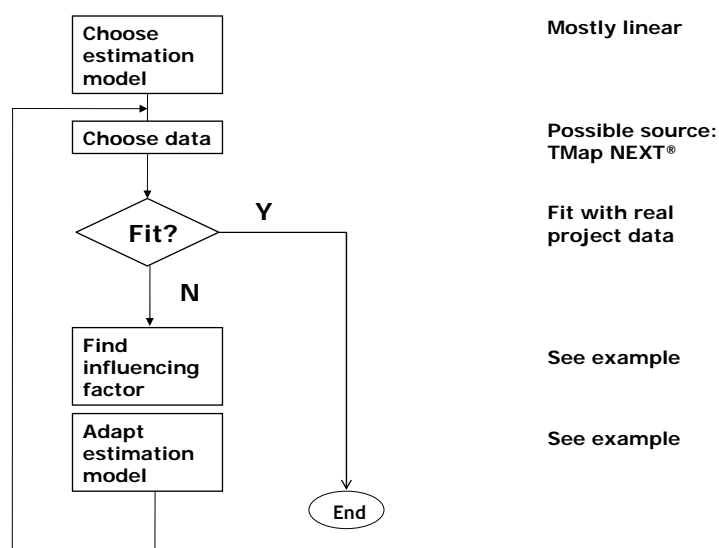
## Processes around metrics

- Collection in a project
  - > Embedded in daily work
  - > Weekly summarisation
  - > Sanity checks
  - > Cost: about 2% project budget
- Distribution of benchmark data
- For a benchmark on the level of:
  - > Project releases
  - > Organisation
  - > Country
  - > International: [www.ISBSG.org](http://www.ISBSG.org)  
International Software Benchmarking Standards Group

## Some Considerations for future use

1. Accuracy of definitions
2. Number of types of defects
3. Is a batch test case the same as an online test case?
4. Only testing of functionality or also security, performance, usability
5. How to include regression testing?
6. Measure personal productivity?
7. Predictive value  
average (mean), median, standard deviation, correlations with?  
Prediction model needed?

## Find a typical model

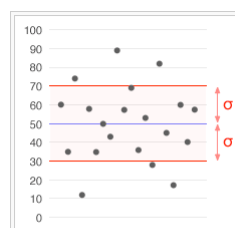


## 10 similar projects

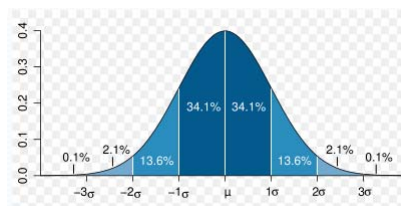
Project	Func Design	Construct	System test	Function Points	FD-hrs per fp	Constr-hrs per fp	Systemtest-hrs per fp
1							1,93
2							2,28
3							2,74
4							3,25
5							3,30
6							3,52
7							4,42
8							4,30
9							6,70
10							7,73
Average							4,0

## Standard deviation

Standard deviation =  
the mean root square (RMS)  
deviation of the values  
from their mean(=average):  $\sigma$



A data set with a mean of 50 and a standard deviation ( $\sigma$ ) of 20



From wikipedia



## And now

- **Non-GIS**
  - > Average = 3.1
  - > Standard deviation = 0.8
  - > As predictor: 68% within 2.3 and 3.9
- **GIS**
  - > Average = 6.2
  - > Standard deviation = 1.4
  - > As predictor: 68% within 4.8 and 7.6
- **Overall**
  - > Average = 4.0
  - > Standard deviation = 1.8
  - > As predictor: 68% within 2.2 and 5.8

## To apply test metrics

- **Project**
  - > Estimating Testing New development
  - > Estimating Testing Maintenance release
  - > Controlling the test project
- **Benchmark**
  - > Measure the effectiveness of the testing, Improve inside / outside the test process?
- **Make a judgement?**

## Make a judgement based on metrics?

- **Of personal production and / or quality of work**
  - > **Some Dutch effects**
- **Of Service Level Agreement**
  - > **What to measure and how to judge**

## Agenda

- **Introduction**
- **Theory**
- **Experience: metrics in practice**
- **How to use, pitfalls**
- **More practical data**
- **Wrap up**
- **Bonus: value of testing**



## Test estimation

- **Early:  
Based on ratio**

Functional design	Realisation	Functional testing	Project type
2	5	3	Waterfall, 3GL, QA
1	3	3	Thin FD, dominant developers
1	2	1,2	Waterfall, test budget max.

## Test estimation

- **After design : Based on object size**

Object type	In TOSM
Function point	1
Data model	35*
# pages requirement	15
# pages functional	3
# Screens	8
Kilo SLOC	6.6 - 60

TOSM = Test Object Size Meter  
Test hours: 1,5-4 hours for 1 TOSM

## Rules of thumb / experience

- **Defect introduction**
  - > 1 if 6 repaired is "market"
  - > 1 if 3 repaired = bad
  - > 3 if 1 repaired is instability
- **Strategy effect**
  - > Ration spent hours testing risk class  
A(high) B(medium) and C (low) :  
4.5            3                    2
- **Lead time measurement is very few seen**

## Agenda

- Introduction
- Theory
- Experience: metrics in practice
- How to use, pitfalls
- More practical data
- Wrap up
- Bonus: Value of testing



## Wrap up

- Metrics are possible
- Metrics are useful
- Be careful in using

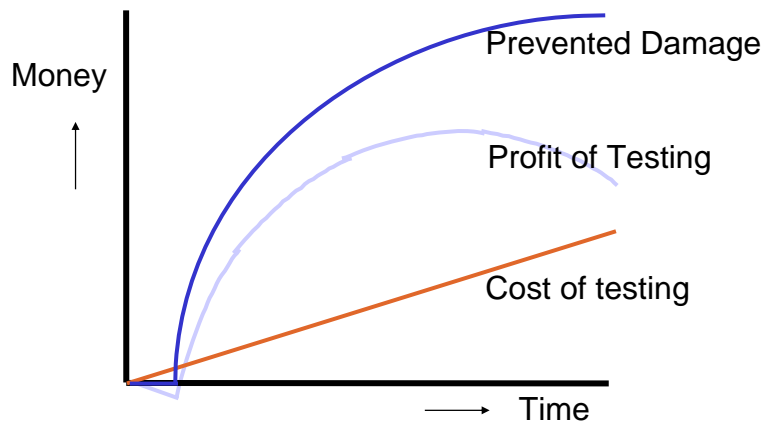
**! Thank you!**  
**? Questions  
or Bonus?**

## What does testing deliver?

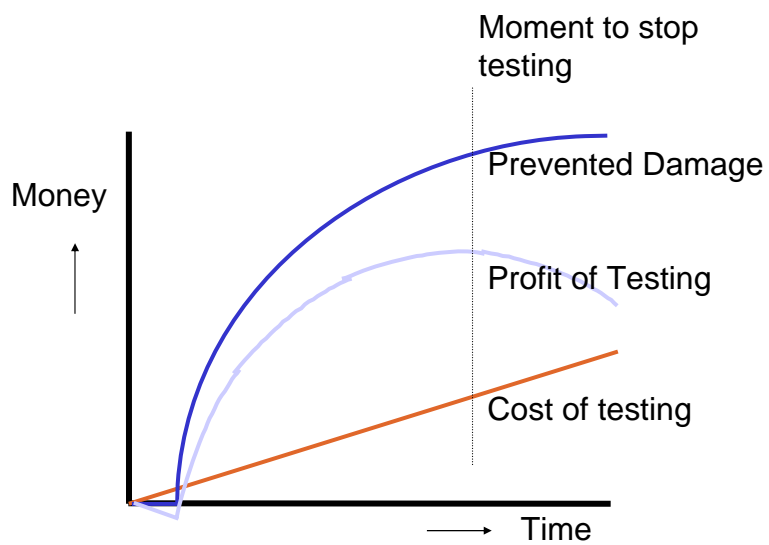
- **Prevention of damage**
  - > Loss of revenue, reputation, productivity
  - > Compensation claims
  - > 'Known errors'
- **Trust in the product**
  - > Quality information of the product
  - > Insight into the risks left
  - > Being tested gives a good feeling
- **Facilitating project management**
  - > Progress and quality information

## Profit of Testing

Profit = Prevented Damage -/- Cost of Testing

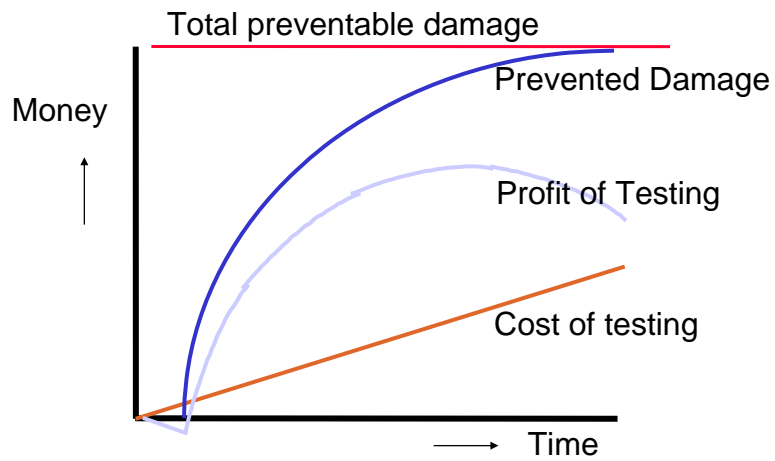


## Moment to stop testing

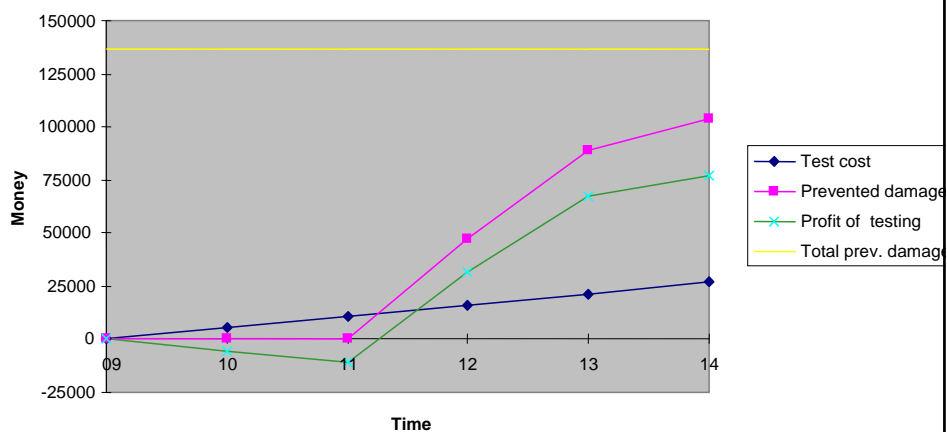


## Total preventable damage

Total preventable damage = Prevented damage in test  
+ Damage in Production (3 months)

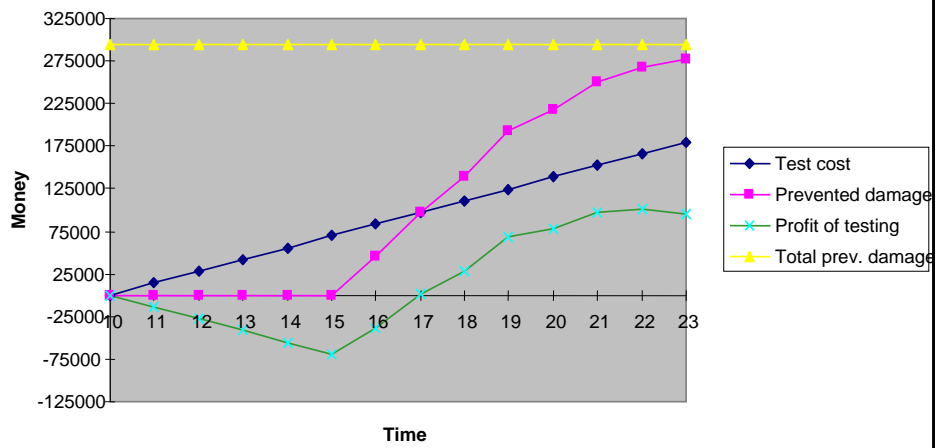


## Experience 1



**Stopped too early!**

## Experience 2



**Stopped in time!**