Measuring the Value of Testing

Prepared and presented by

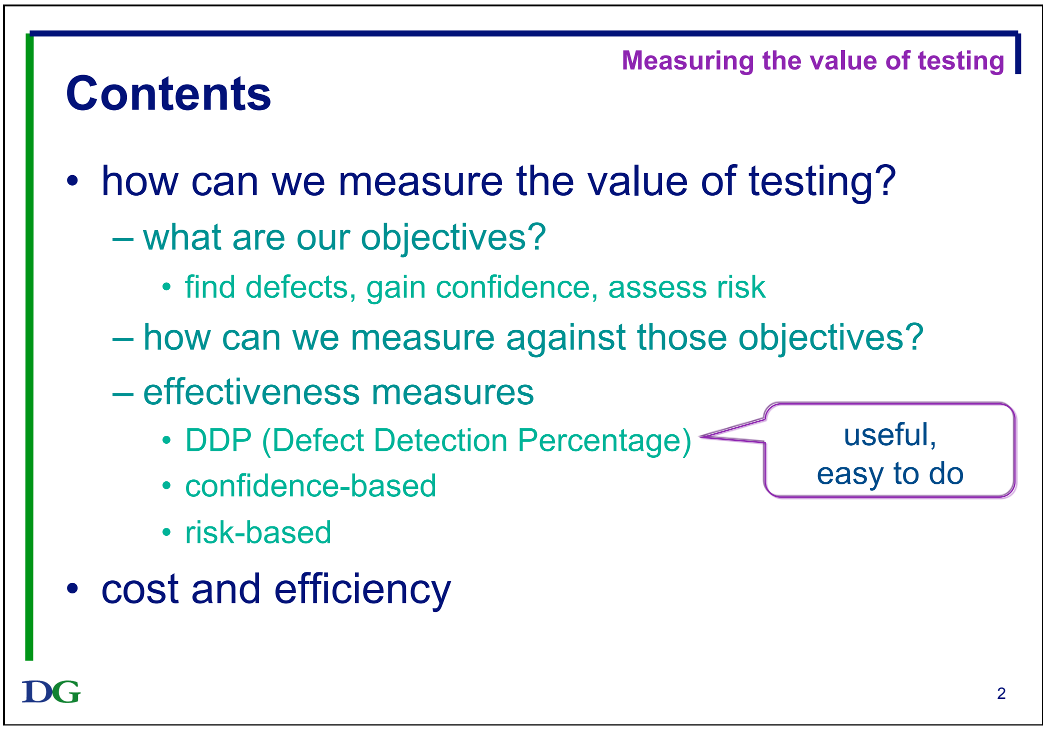
Dorothy Graham

email: info@dorothygraham.co.uk

www.DorothyGraham.co.uk

© Dorothy Graham 2009

DG



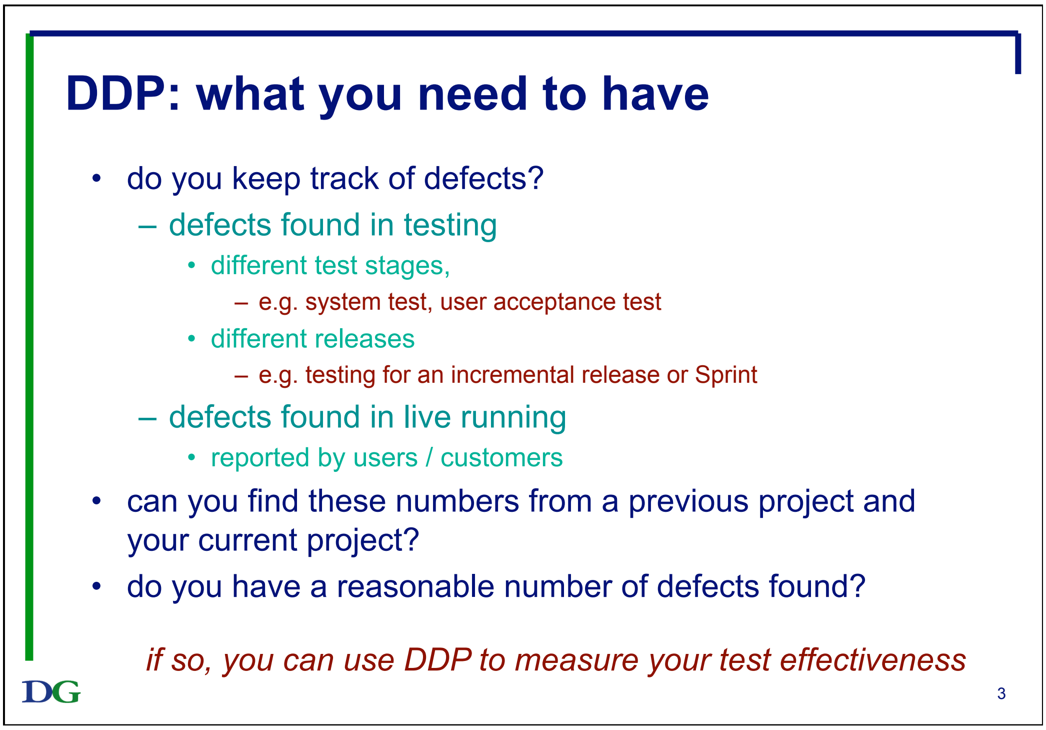
Measuring the value of testing

Contents

- how can we measure the value of testing?
  - what are our objectives?
    - find defects, gain confidence, assess risk
  - how can we measure against those objectives?
  - effectiveness measures
    - DDP (Defect Detection Percentage)
    - confidence-based
    - risk-based
- cost and efficiency

useful,
easy to do

DG

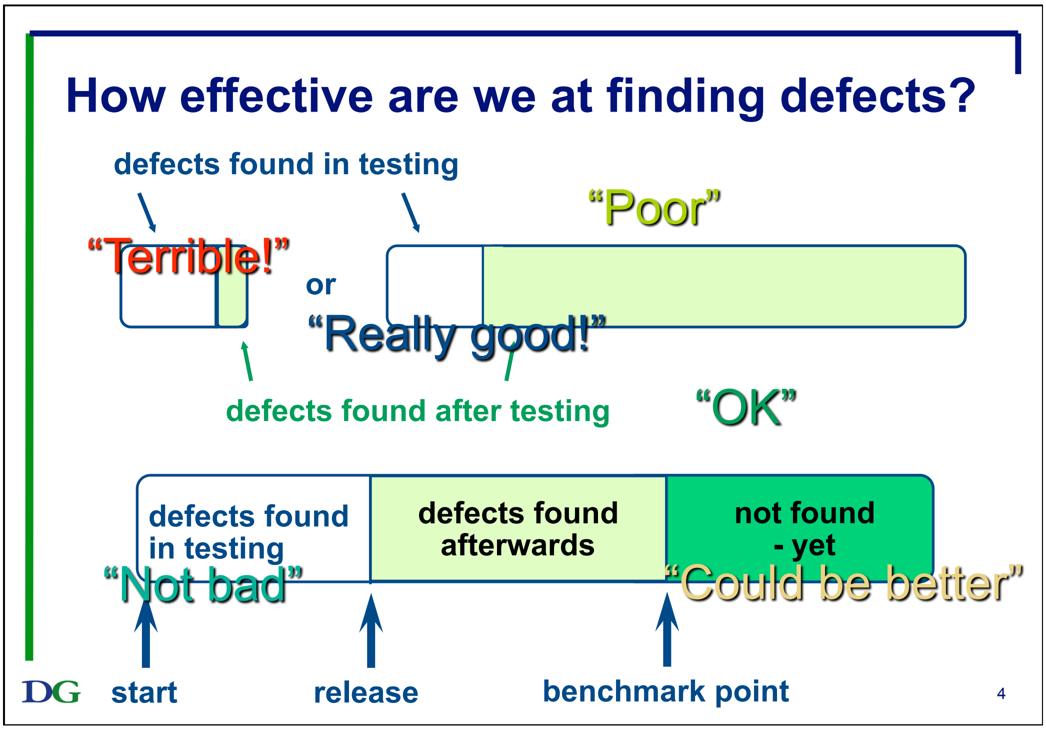


DDP: what you need to have

- do you keep track of defects?
  - defects found in testing
    - different test stages,
      - e.g. system test, user acceptance test
    - different releases
      - e.g. testing for an incremental release or Sprint
  - defects found in live running
    - reported by users / customers
- can you find these numbers from a previous project and your current project?
- do you have a reasonable number of defects found?

if so, you can use DDP to measure your test effectiveness

DG



How effective are we at finding defects?

defects found in testing

"Poor"

"Terrible!"

or

"Really good!"

defects found after testing

"OK"

defects found in testing

defects found afterwards

not found - yet

"Not bad"

"Could be better"

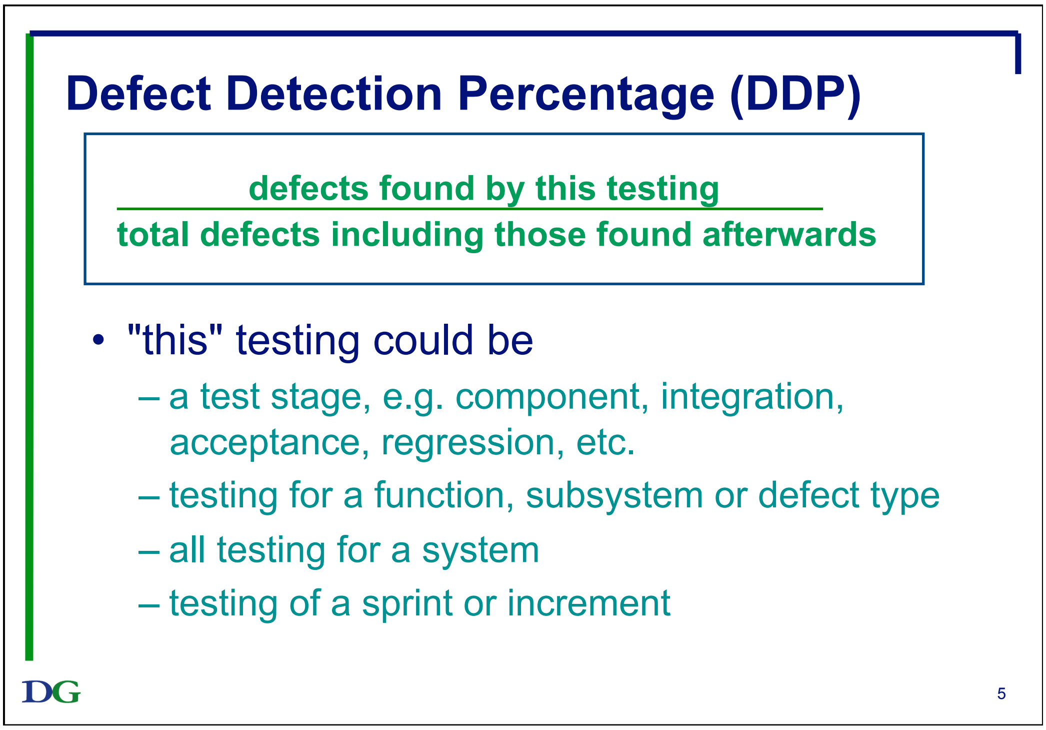
start

release

benchmark point

DG

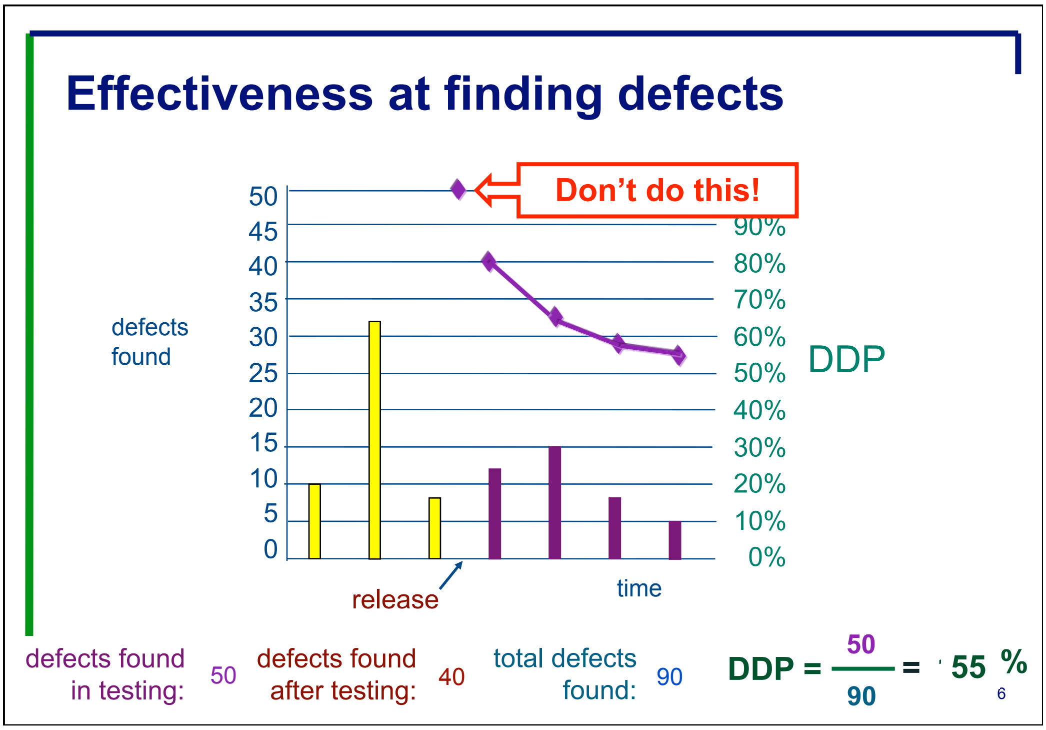
© Dorothy Graham 2009



Defect Detection Percentage (DDP)

$$\frac{\text{defects found by this testing}}{\text{total defects including those found afterwards}}$$

- "this" testing could be
  - a test stage, e.g. component, integration, acceptance, regression, etc.
  - testing for a function, subsystem or defect type
  - all testing for a system
  - testing of a sprint or increment



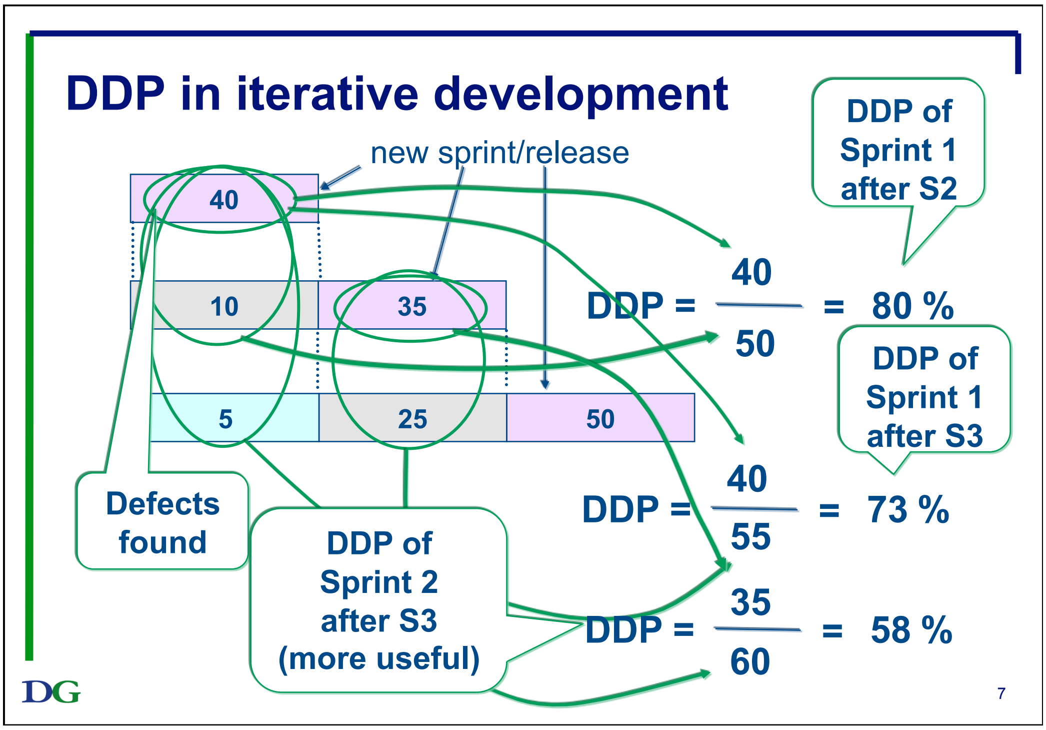
Effectiveness at finding defects

Don't do this!

defects found in testing: 50 defects found after testing: 40 total defects found: 90

$$\text{DDP} = \frac{50}{90} = \text{'}\ 55\ \%$$

© Dorothy Graham 2009
www.DorothyGraham.co.uk



DDP in iterative development

DDP of Sprint 1 after S2

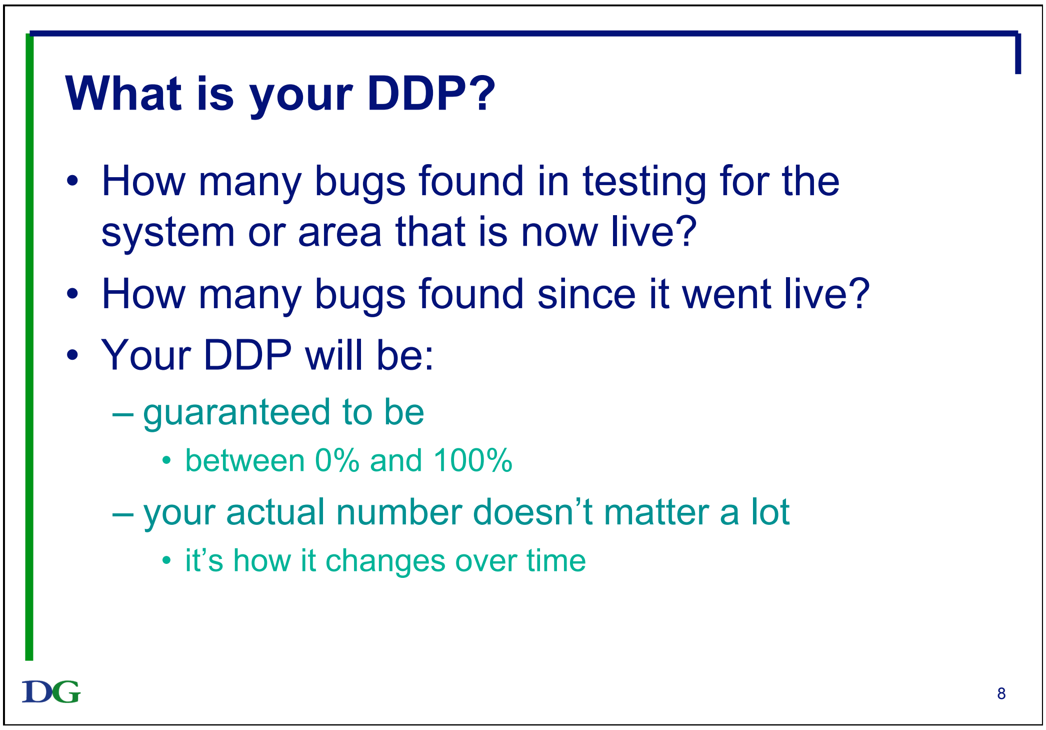
$$\text{DDP} = \frac{40}{50} = 80\ \%$$

DDP of Sprint 1 after S3

$$\text{DDP} = \frac{40}{55} = 73\ \%$$

$$\text{DDP} = \frac{35}{60} = 58\ \%$$

DG

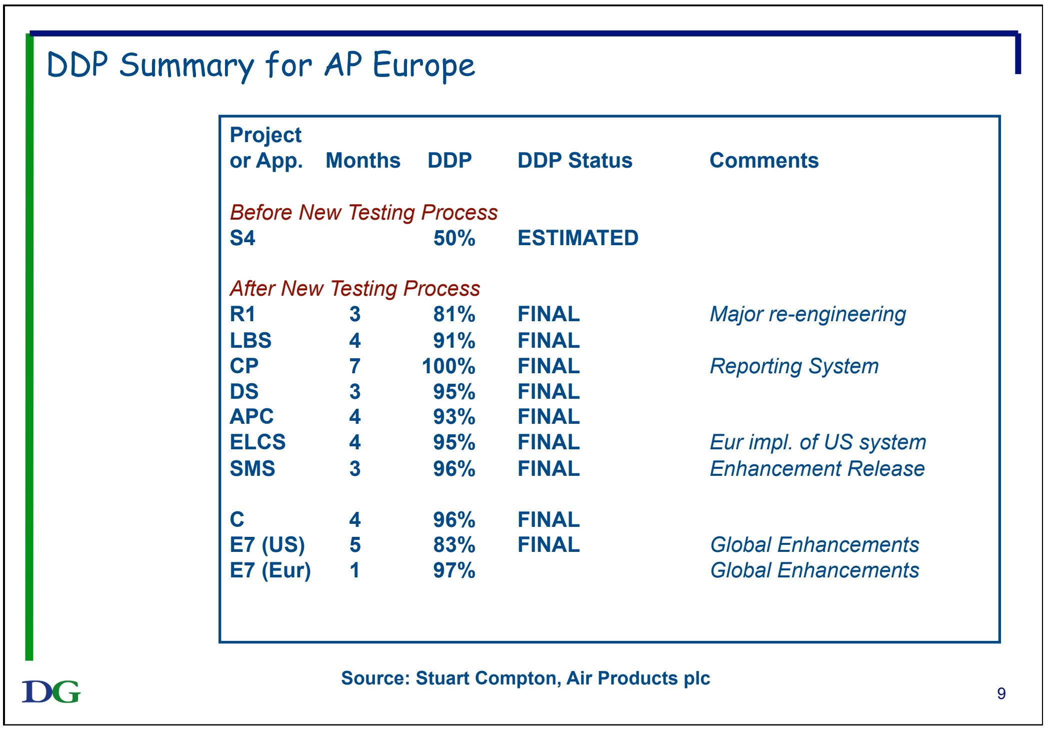


What is your DDP?

- How many bugs found in testing for the system or area that is now live?
- How many bugs found since it went live?
- Your DDP will be:
  - guaranteed to be
    - between 0% and 100%
  - your actual number doesn't matter a lot
    - it's how it changes over time

DG

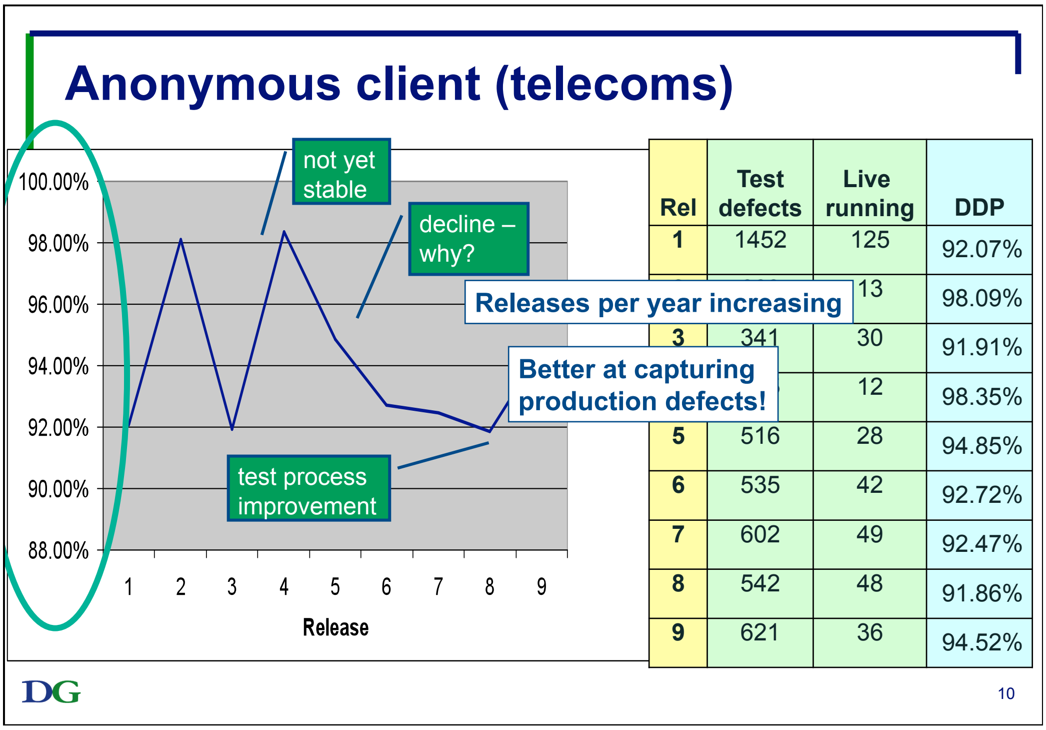
© Dorothy Graham 2009



DDP Summary for AP Europe

Project or App.	Months	DDP	DDP Status	Comments
Before New Testing Process				
S4		50%	ESTIMATED	
After New Testing Process				
R1	3	81%	FINAL	*Major re-engineering*
LBS	4	91%	FINAL	
CP	7	100%	FINAL	*Reporting System*
DS	3	95%	FINAL	
APC	4	93%	FINAL	
ELCS	4	95%	FINAL	*Eur impl. of US system*
SMS	3	96%	FINAL	*Enhancement Release*
C	4	96%	FINAL	
E7 (US)	5	83%	FINAL	*Global Enhancements*
E7 (Eur)	1	97%		*Global Enhancements*

Source: Stuart Compton, Air Products plc

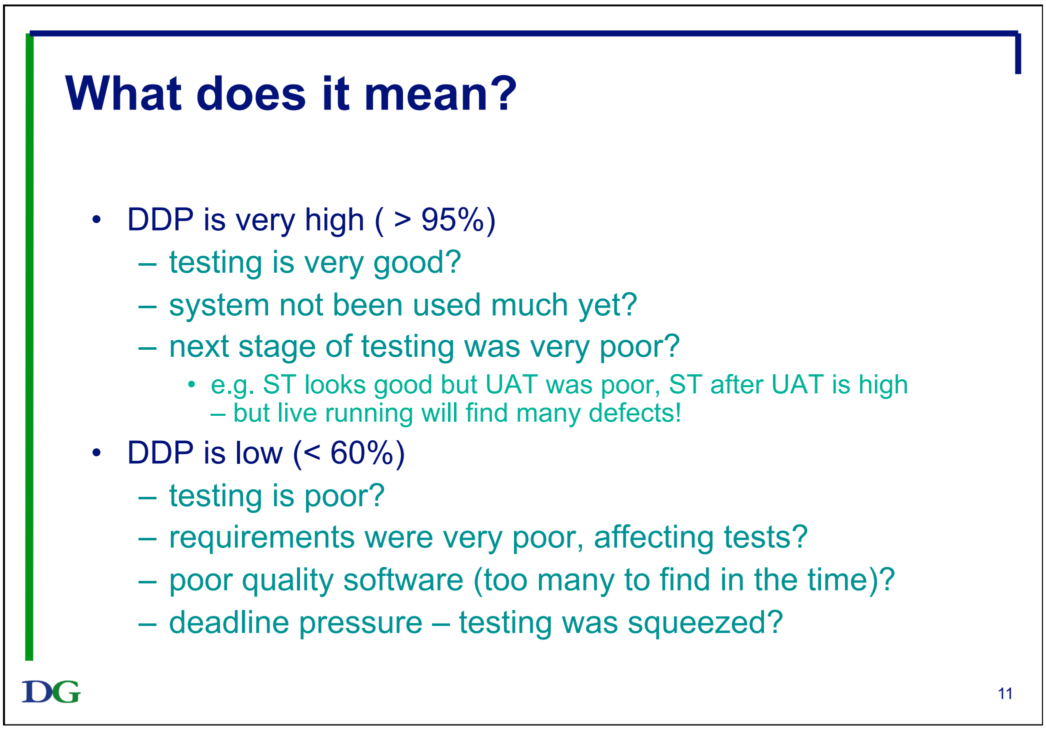


Anonymous client (telecoms)

Rel	Test defects	Live running	DDP
1	1452	125	92.07%
[illegible]	[illegible]	13	98.09%
3	341	30	91.91%
[illegible]	[illegible]	12	98.35%
5	516	28	94.85%
6	535	42	92.72%
7	602	49	92.47%
8	542	48	91.86%
9	621	36	94.52%

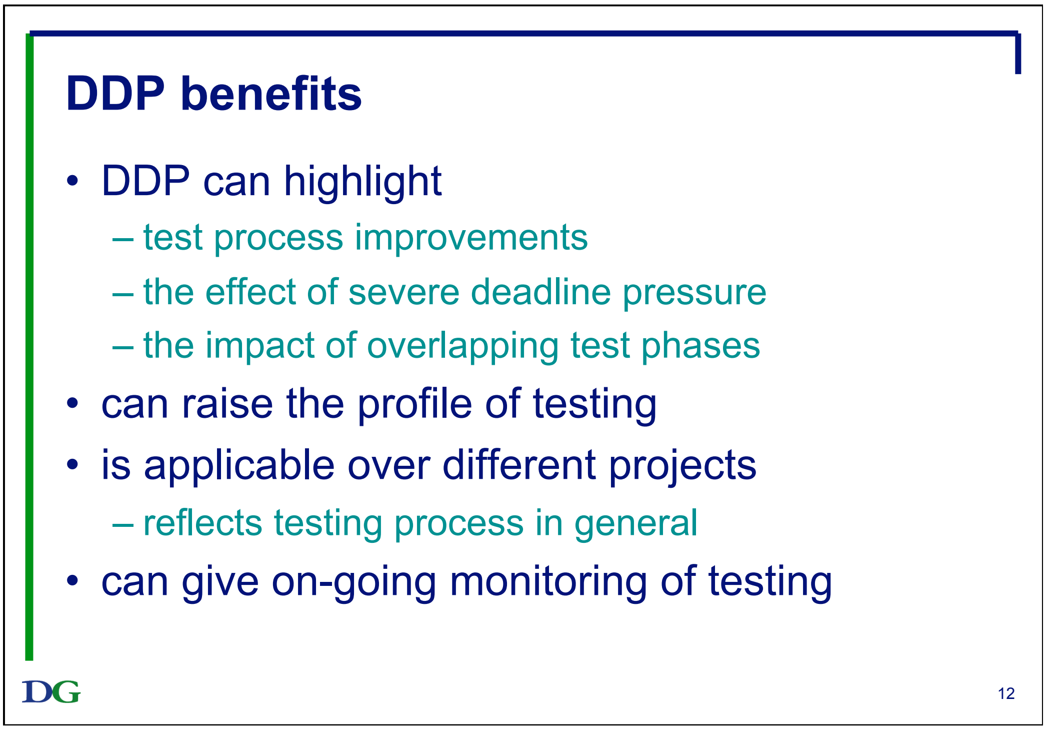
Releases per year increasing

Better at capturing production defects!



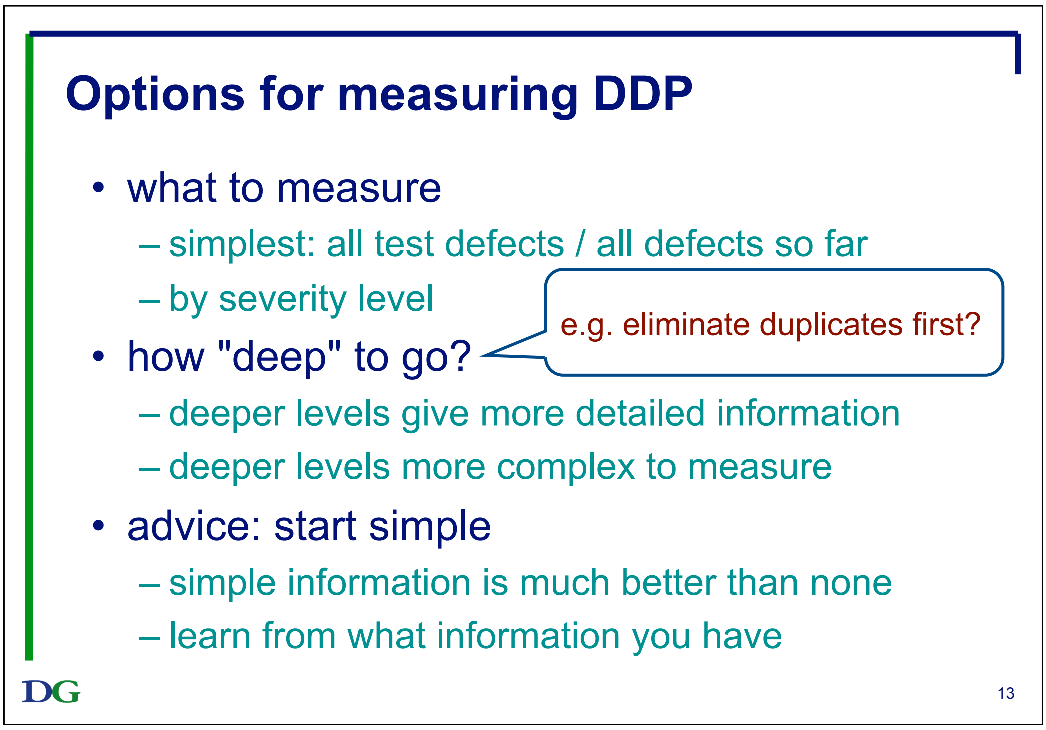
What does it mean?

- DDP is very high (> 95%)
  - testing is very good?
  - system not been used much yet?
  - next stage of testing was very poor?
    - e.g. ST looks good but UAT was poor, ST after UAT is high – but live running will find many defects!
- DDP is low (< 60%)
  - testing is poor?
  - requirements were very poor, affecting tests?
  - poor quality software (too many to find in the time)?
  - deadline pressure – testing was squeezed?



DDP benefits

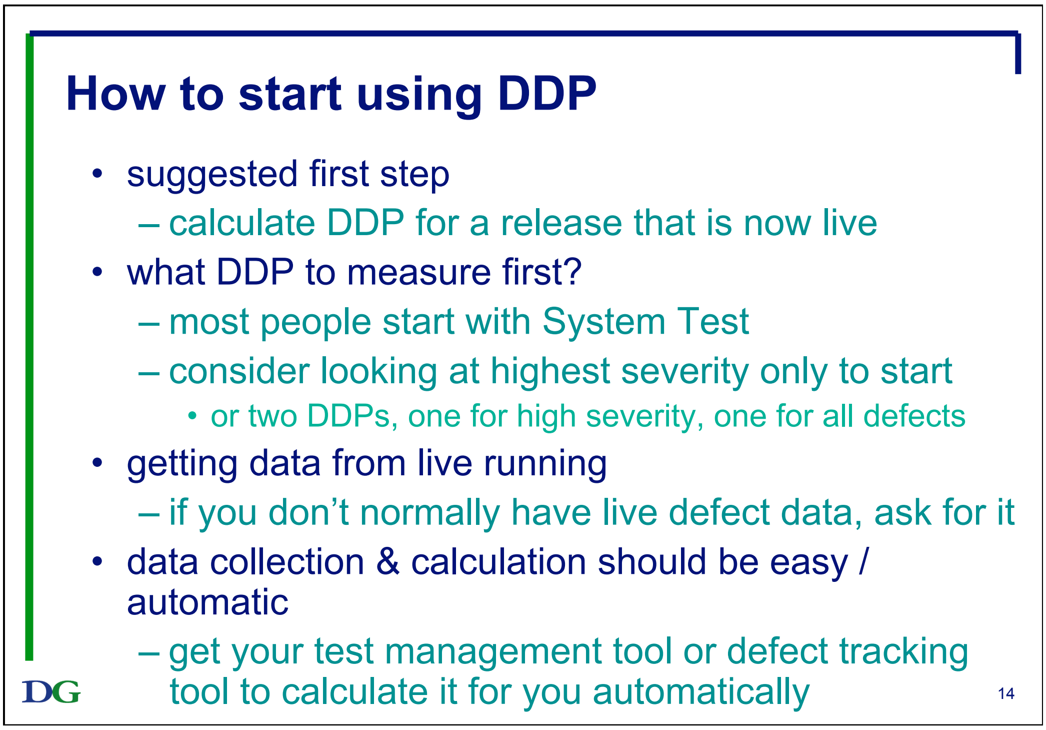
- DDP can highlight
  - test process improvements
  - the effect of severe deadline pressure
  - the impact of overlapping test phases
- can raise the profile of testing
- is applicable over different projects
  - reflects testing process in general
- can give on-going monitoring of testing



Options for measuring DDP

- what to measure
  - simplest: all test defects / all defects so far
  - by severity level
- how "deep" to go?
  - deeper levels give more detailed information
  - deeper levels more complex to measure
- advice: start simple
  - simple information is much better than none
  - learn from what information you have

e.g. eliminate duplicates first?



How to start using DDP

- suggested first step
  - calculate DDP for a release that is now live
- what DDP to measure first?
  - most people start with System Test
  - consider looking at highest severity only to start
    - or two DDPs, one for high severity, one for all defects
- getting data from live running
  - if you don’t normally have live defect data, ask for it
- data collection & calculation should be easy / automatic
  - get your test management tool or defect tracking tool to calculate it for you automatically

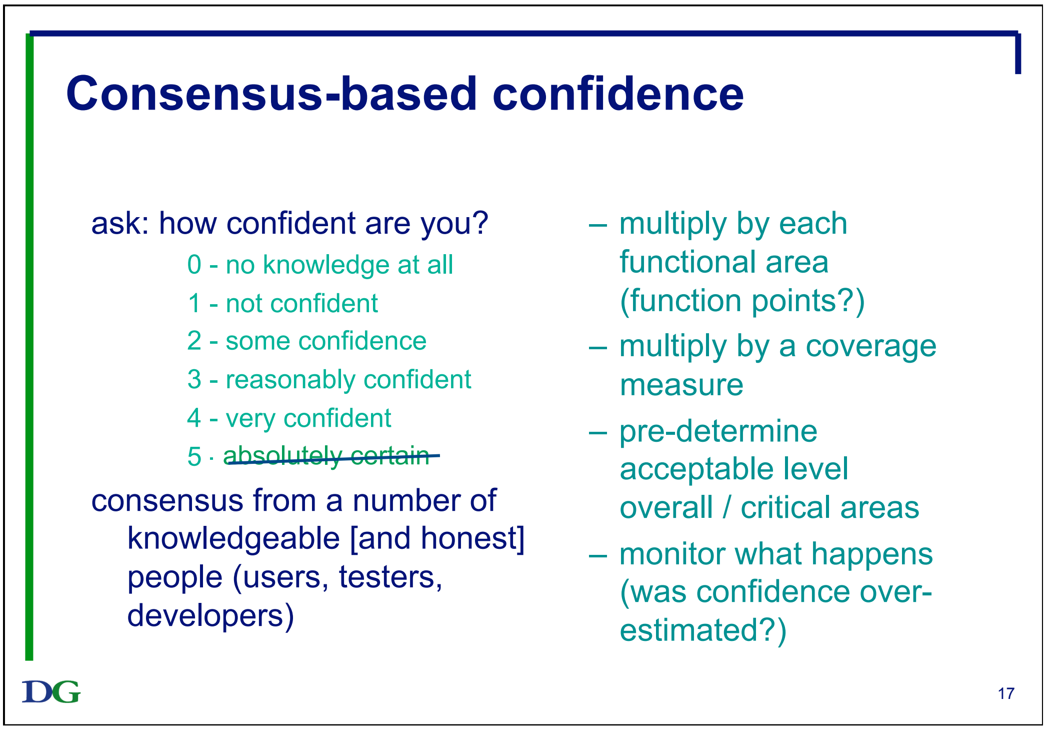
Contents

- how can we measure the value of testing?
  - what are our objectives?
  - how can we measure against those objectives?
  - effectiveness measures
    - DDP
    - confidence-based
    - risk-based
- cost and efficiency

Confident about what?

- the system being tested
  - will be usable - will meet a business need
  - will do the right things
  - will be reliable (not fail in operation)
- testing has been adequate
  - right areas tested
  - right depth of testing for critical / non-critical areas
  - testing done correctly (process followed)
- ok to release?

© Dorothy Graham 2009



Consensus-based confidence

ask: how confident are you?

- 0 - no knowledge at all
- 1 - not confident
- 2 - some confidence
- 3 - reasonably confident
- 4 - very confident
- 5 - ~~absolutely certain~~

consensus from a number of knowledgeable [and honest] people (users, testers, developers)

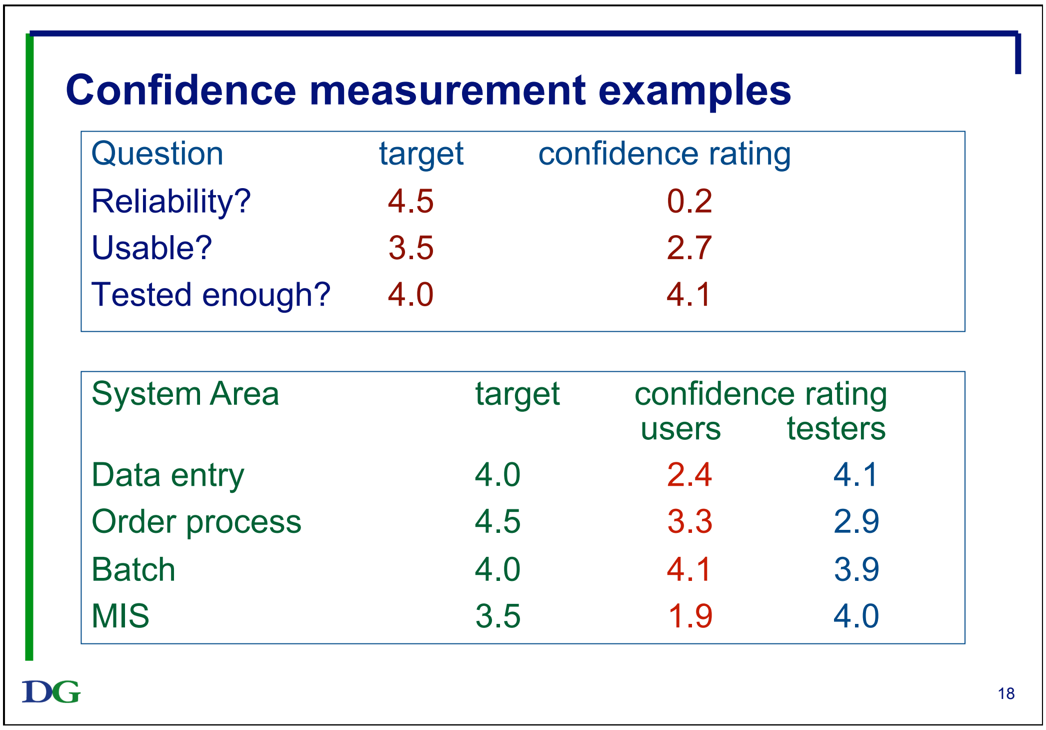
– multiply by each functional area (function points?)

– multiply by a coverage measure

– pre-determine acceptable level overall / critical areas

– monitor what happens (was confidence over-estimated?)

DG



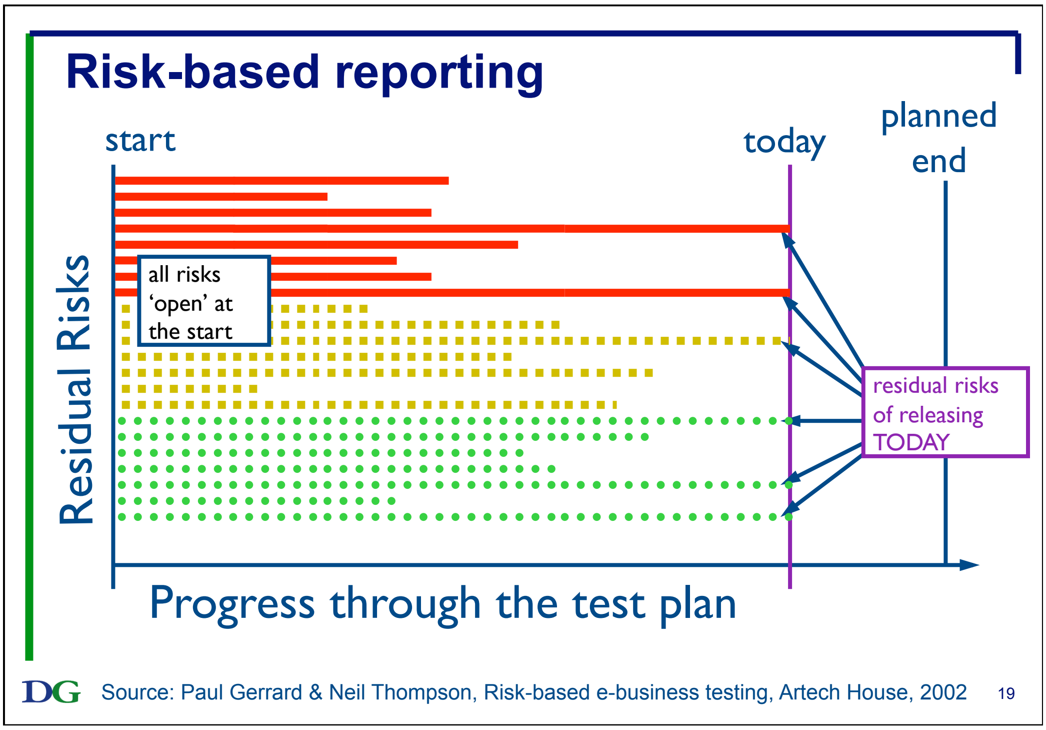
Confidence measurement examples

Question	target	confidence rating
Reliability?	4.5	0.2
Usable?	3.5	2.7
Tested enough?	4.0	4.1

System Area	target	confidence rating	
		users	testers
Data entry	4.0	2.4	4.1
Order process	4.5	3.3	2.9
Batch	4.0	4.1	3.9
MIS	3.5	1.9	4.0

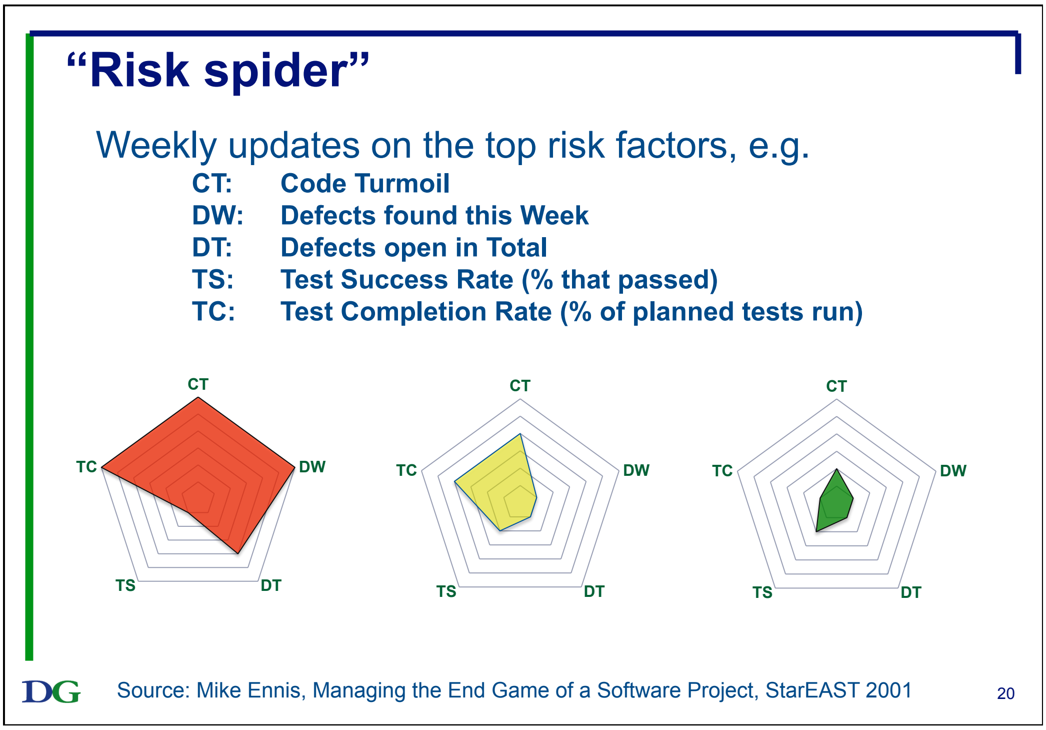
DG

© Dorothy Graham 2009



Risk-based reporting

Source: Paul Gerrard & Neil Thompson, Risk-based e-business testing, Artech House, 2002



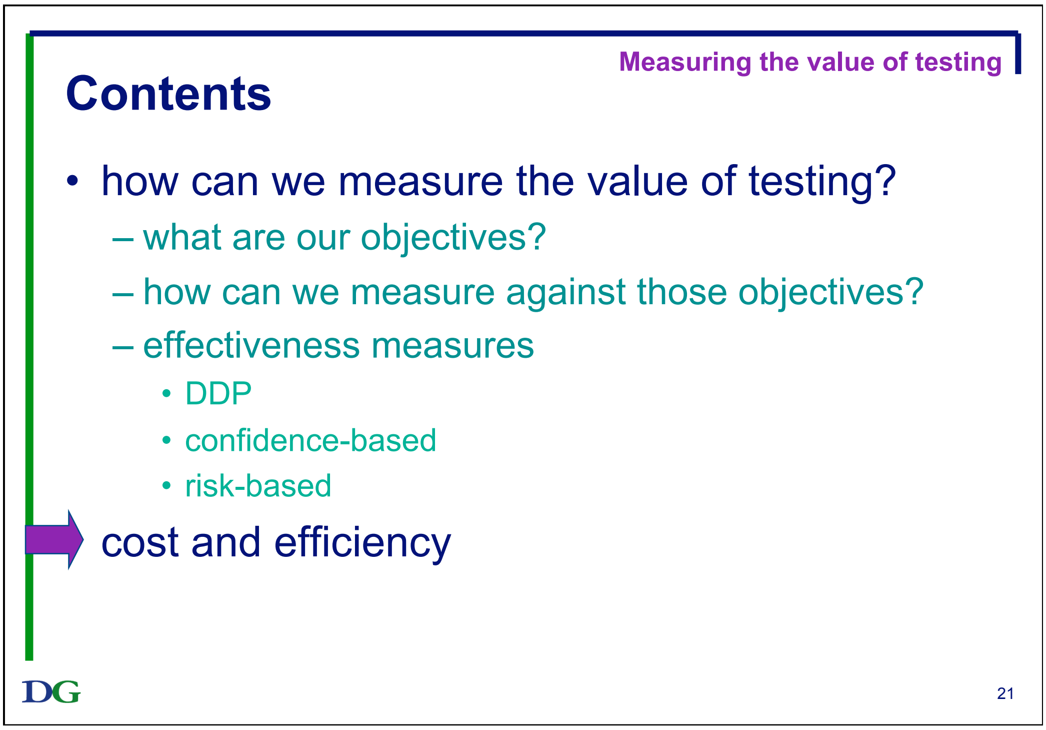
"Risk spider"

Weekly updates on the top risk factors, e.g.

CT:	**Code Turmoil**
DW:	**Defects found this Week**
DT:	**Defects open in Total**
TS:	**Test Success Rate (% that passed)**
TC:	**Test Completion Rate (% of planned tests run)**

Source: Mike Ennis, Managing the End Game of a Software Project, StarEAST 2001

© Dorothy Graham 2009



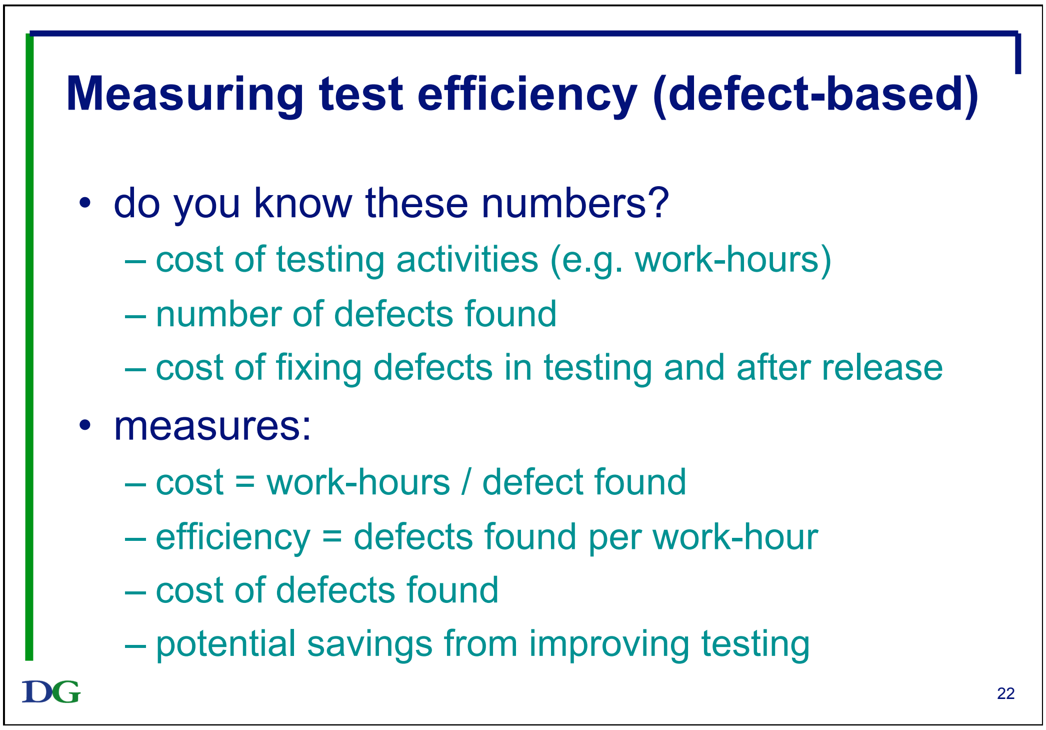
Measuring the value of testing

Contents

- how can we measure the value of testing?
  - what are our objectives?
  - how can we measure against those objectives?
  - effectiveness measures
    - DDP
    - confidence-based
    - risk-based

➡ cost and efficiency

DG



Measuring test efficiency (defect-based)

- do you know these numbers?
  - cost of testing activities (e.g. work-hours)
  - number of defects found
  - cost of fixing defects in testing and after release
- measures:
  - cost = work-hours / defect found
  - efficiency = defects found per work-hour
  - cost of defects found
  - potential savings from improving testing

DG

© Dorothy Graham 2009

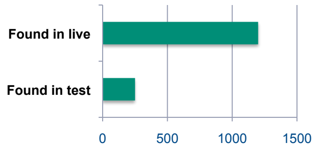
Measuring cost & efficiency – example 1

- 100 hours testing effort, found 20 defects
  - cost: 5 hrs/defect
  - efficiency: 0.2 defects/hour
- 100 hours testing effort, found 200 defects
  - cost: 0.5 hrs/defect
  - efficiency: 2 defects/hour
- which is better?
  - not enough information!

DG

Measuring test cost – example 2

Cost per bug



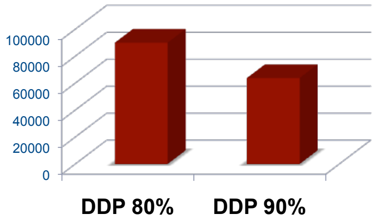
in test: $150, in live: $1200

DDP = 80%

if 200 found in test, 50 in live,

Total cost = **$90,000**

Total cost



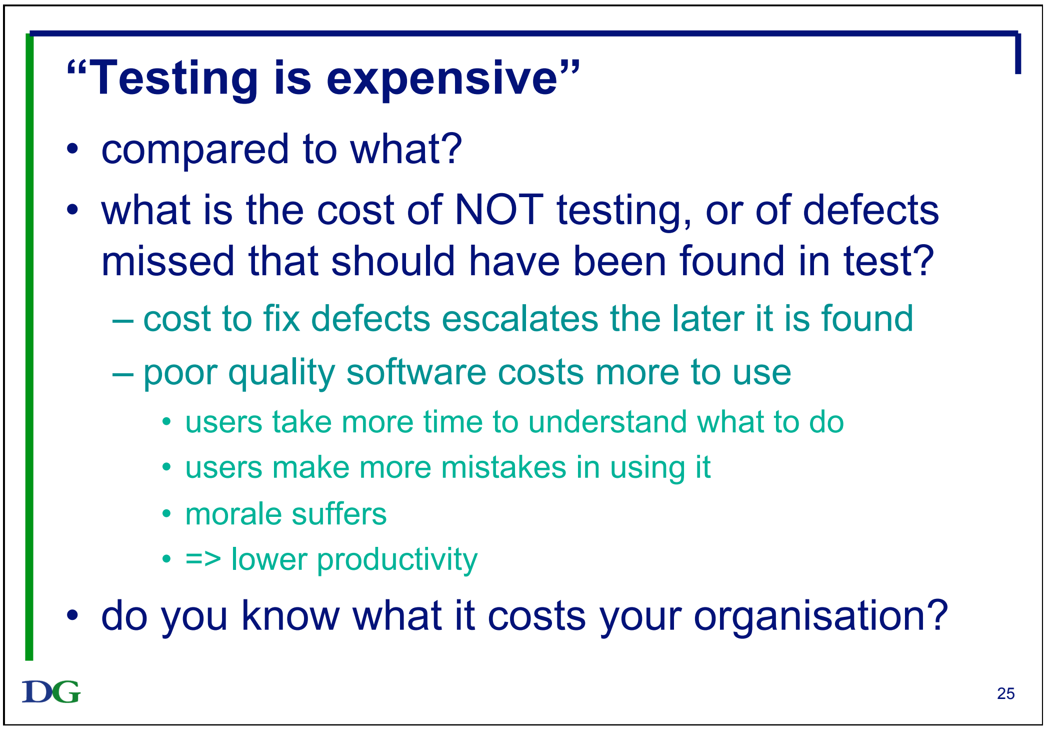
what if DDP was 90%?
225 found in test, 25 in live,

Total cost = **$63,750**

increasing DDP by 10% would save $26,250

DG

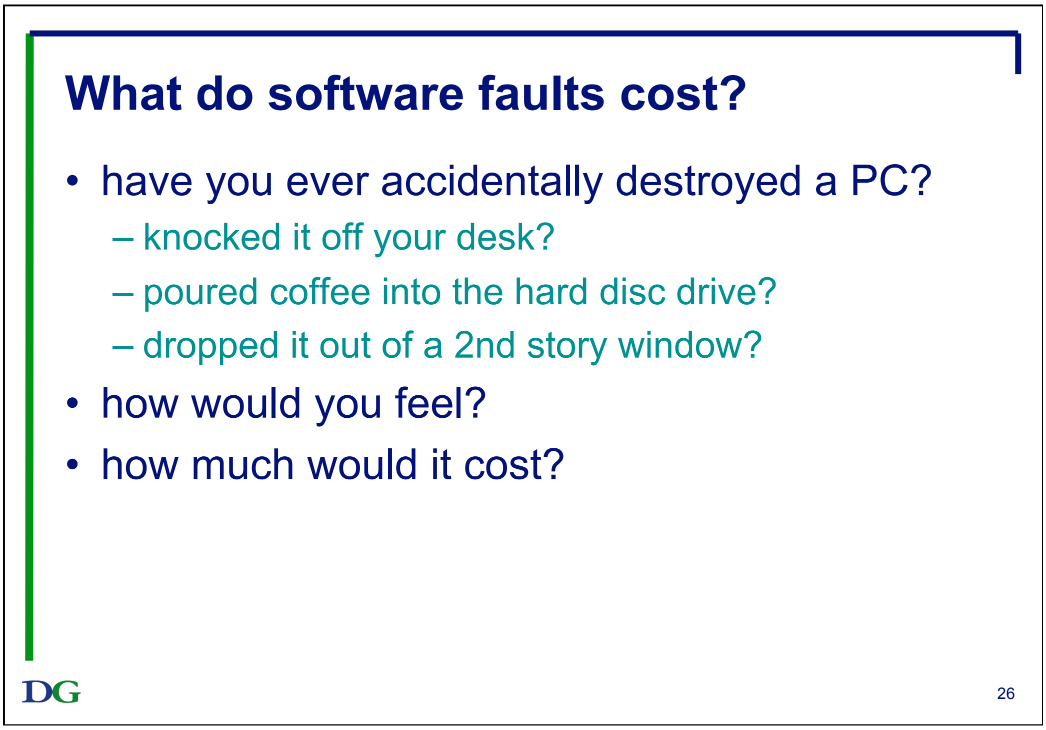
© Dorothy Graham 2009



“Testing is expensive”

- compared to what?
- what is the cost of NOT testing, or of defects missed that should have been found in test?
  - – cost to fix defects escalates the later it is found
  - – poor quality software costs more to use
    - users take more time to understand what to do
    - users make more mistakes in using it
    - morale suffers
    - => lower productivity
- do you know what it costs your organisation?

25



What do software faults cost?

- have you ever accidentally destroyed a PC?
  - – knocked it off your desk?
  - – poured coffee into the hard disc drive?
  - – dropped it out of a 2nd story window?
- how would you feel?
- how much would it cost?

26

Hypothetical Cost - 1

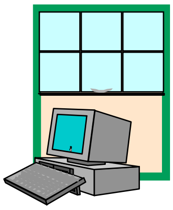
(Loaded Salary cost: 500/hr)

Cost	Developer	User
- detect (.5 hr)		250
- report (.5 hr)		250
- receive & process (1 hr)	500	
- assign & bkgnd (4 hrs)	2000	
- debug (.5 hr)	250	
- test bug fix (.5 hr)	250	
- regression test (8 hrs)	4000	
	7000	500

DG

Hypothetical Cost - 2

Cost	Developer	User
	7000	500
- update doc'n, CM (2 hrs)	1000	
- update code library (1 hr)	500	
- inform users (1 hr)	500	
- admin(10% = 2 hrs)	1000	
Total (20 hrs)	10.000	



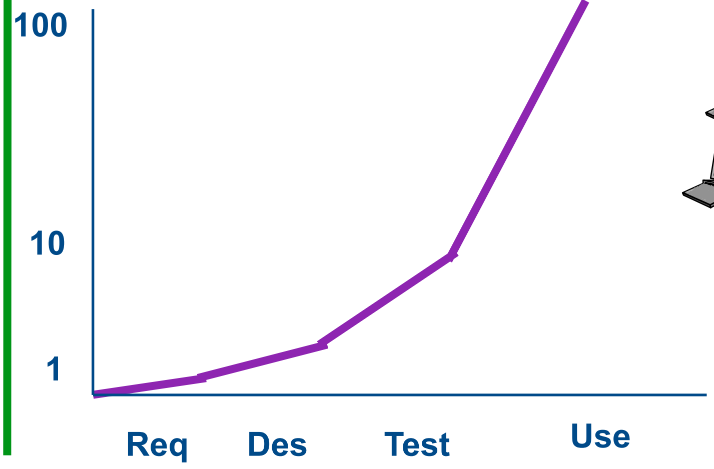
DG

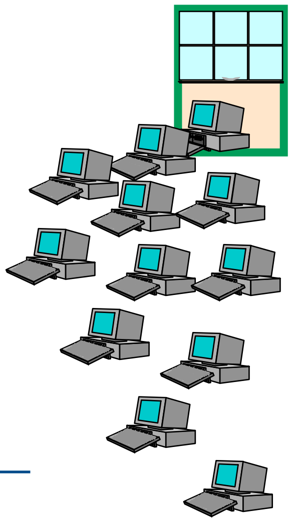
© Dorothy Graham 2009

Hypothetical Cost - 3

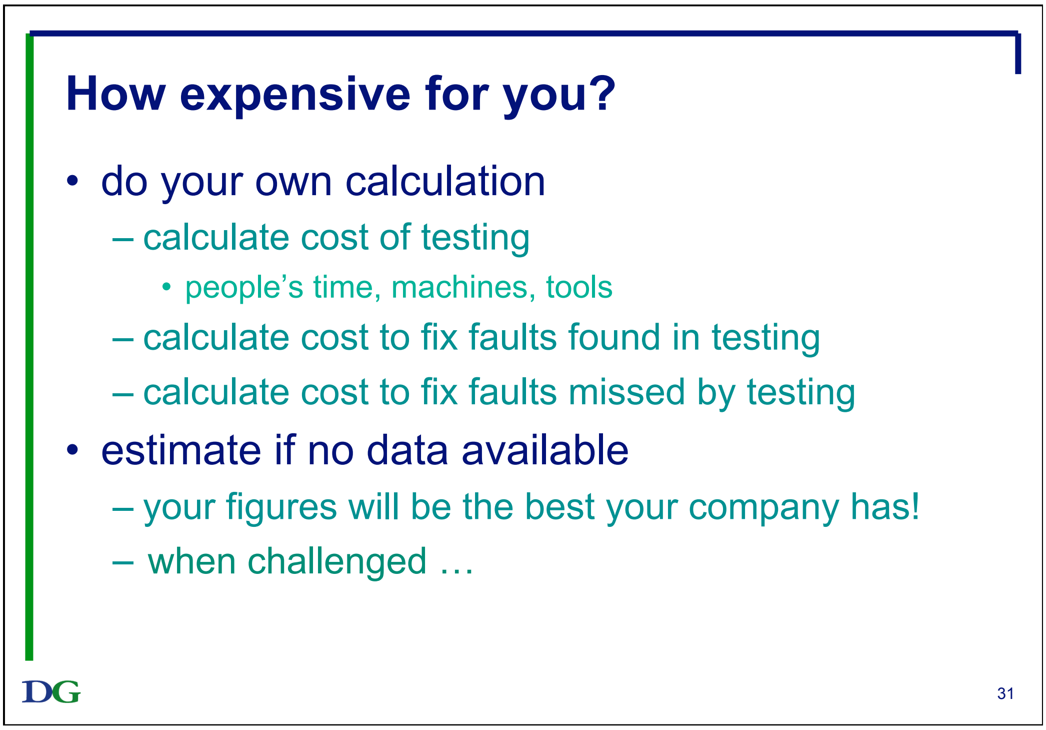
Cost	Developer	User
	10.000	500
(suppose affects only 5 users)		
- work x 2, 1 wk		40000
- fix data (1 day)		3500
- pay for fix (3 days maint)		7500
- regr test & sign off (2 days)		7000
- update doc'n / inform (1 day)		3500
- double check + 12% 5 wks		50000
- admin (+7.5%)		8000
Totals	10.000	120.000

Cost of fixing defects





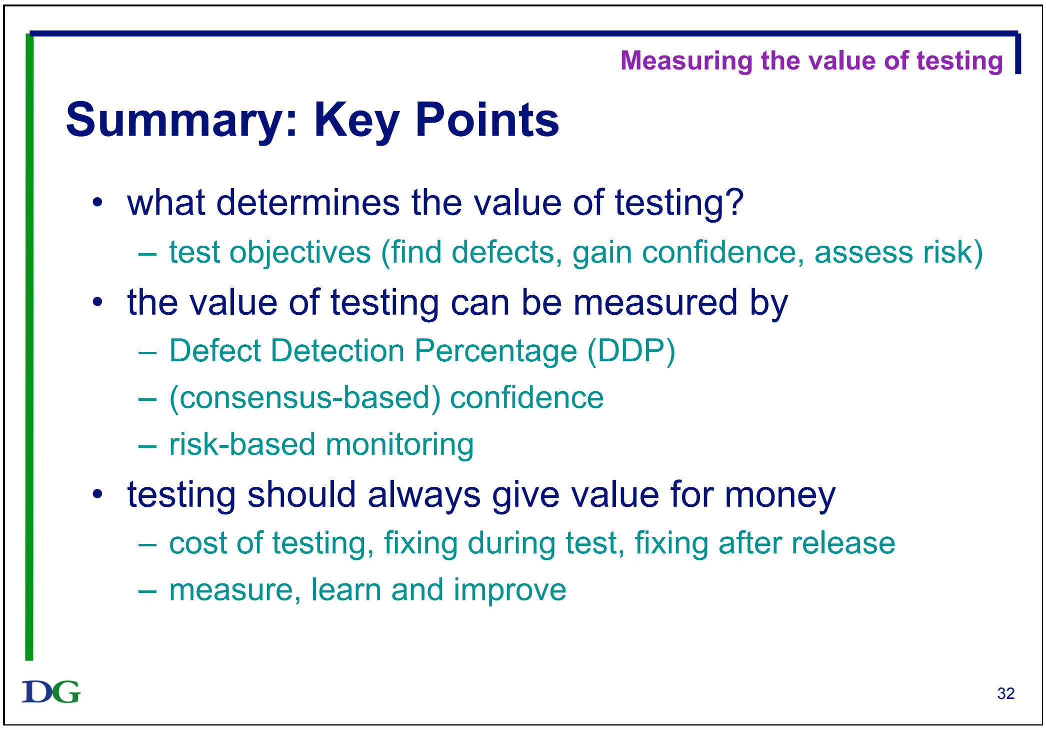
© Dorothy Graham 2009



How expensive for you?

- do your own calculation
  - calculate cost of testing
    - people’s time, machines, tools
  - calculate cost to fix faults found in testing
  - calculate cost to fix faults missed by testing
- estimate if no data available
  - your figures will be the best your company has!
  - when challenged …

DG



Summary: Key Points

- what determines the value of testing?
  - test objectives (find defects, gain confidence, assess risk)
- the value of testing can be measured by
  - Defect Detection Percentage (DDP)
  - (consensus-based) confidence
  - risk-based monitoring
- testing should always give value for money
  - cost of testing, fixing during test, fixing after release
  - measure, learn and improve

DG

More information on DDP (slides & exercises)

- download from www.DorothyGraham.co.uk

- discussion on my blog:
  http://dorothygraham.blogspot.com

Further questions / comments:
info@dorothygraham.co.uk