MEASURING TESTING EFFECTIVENESS USING DEFECT DETECTION PERCENTAGE

Dorothy Graham
Grove Consultants UK
Dorothy Graham

Dorothy Graham is the founder of Grove Consultants in the UK, which provides advice, training and inspiration in software testing, testing tools and Inspection. Originally from Grand Rapids Michigan, she attended Calvin College and Purdue University, married an Englishman and has lived and worked in the UK for over 25 years.


Dorothy was Programme Chair for the first EuroSTAR Conference in 1993. She is on the boards of conferences and journals in software testing, and has been an active member of the British Computer Society's Specialist Interest Group in Software Testing since 1989, including a working party to produce a new software component testing standard. She was a founder member of the Software Testing Board of the Information Systems Examination Board (ISEB) of the British Computer Society.

She has been awarded the European Excellence Award in Software Testing.

Grove Consultants

Grove Consultants provides consultancy, training and inspiration in software testing. It comprises Dorothy Graham, Mark Fewster, Lloyd Roden and Clive Bates.

Grove Consultants bring a unique perspective to any organisation's software testing processes. Because they specialise exclusively in testing, they have a great depth of knowledge in this field. With a broad range of clients, they bring an industry-wide perspective on testing with a focus on a European perspective. Consultancy assignments have included audits of testing practices, reviews of testing methodologies, assessment of testing effectiveness and efficiency, and advice on all aspects of testing, test automation and Inspection.

Grove Consultants also provide training courses in a range of software testing topics and are the UK's leading accredited training provider for the ISEB Software Testing Foundation and Practitioner Certificates, Europe's first formal qualification for testers.

For more information visit our web site: www.grove.co.uk
Measuring test effectiveness using Defect Detection Percentage (DDP)

Prepared by

Grove Consultants

Contents

- Introduction: some questions for you
- What is DDP
- Case studies
- Examples of calculating DDP
- Advice about DDP & Conclusion
Questions you may be asked

How good is the testing anyway?
Can you prove you are doing a good job?

Your testing can still be just as good in less time, can’t it?
(That deadline pressure really didn’t matter, did it?)

Is the testing any better for this release?
(Have we learned anything?)
(Have we really improved our testing?)

How many bugs have we missed?

Are we better or worse in our testing
compared to other groups/organizations?

Some questions for you

- **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 in RAD
  - 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
Contents

- Introduction: some questions for you
- What is DDP
- Case studies
- Examples of calculating DDP
- Advice about DDP & Conclusion

How effective are we at finding faults?

faults (defects) found in testing

or

faults (defects) found after testing

Defects found in testing  Defects found afterwards  Not found - yet

Start  Release  Benchmark point
Defect Detection Percentage (DDP)

<table>
<thead>
<tr>
<th>Defects found by this testing</th>
<th>Total defects including those found afterwards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defects found</td>
<td>Total defects found</td>
</tr>
</tbody>
</table>

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

Effectiveness at finding defects

\[
DDP = \frac{50}{88} = 57\%
\]
Contents

- Introduction: some questions for you
- What is DDP
- Case studies
  - Examples of calculating DDP
  - Advice about DDP & Conclusion

Case studies from clients

<table>
<thead>
<tr>
<th></th>
<th>Finance (insurance)</th>
<th>Operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 mo</td>
<td>10 mo</td>
</tr>
<tr>
<td>Year 1</td>
<td>70%</td>
<td>50% est</td>
</tr>
<tr>
<td>Year 2</td>
<td>92%</td>
<td></td>
</tr>
</tbody>
</table>

System Test Group DDP = 38%  
(between performance testing)  
Priority 1 & 2 only: DDP = 31%

23% to 87% by application

Defects: 1 / 4    160 / 40

Not useful for low numbers of defects
RESULTS SO FAR

Effectiveness of Testing

Source: Dave Norman, EuroSTAR02 DDP Advanced Workshop, with permission

MESSAGES

Conclusions

- UAT more variable than ST – mainly personnel
- Target zone for ST: 75 - 90%
- Factors behind the figures
  - size, complexity, tester experience, time, documentation
  - whether UAT started before ST was finished
  - where on the S-curve when stopped
- Figures don’t tell you
  - cost, severity of those you missed
  - cost of finding

Page 12
## DDP Summary for AP Europe

<table>
<thead>
<tr>
<th>Project or App.</th>
<th>Months</th>
<th>DDP</th>
<th>DDP Status</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before New Testing Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>50%</td>
<td></td>
<td>ESTIMATED</td>
<td></td>
</tr>
<tr>
<td>After New Testing Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>3</td>
<td>81%</td>
<td>FINAL</td>
<td>Major re-engineering</td>
</tr>
<tr>
<td>LBS</td>
<td>4</td>
<td>91%</td>
<td>FINAL</td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>7</td>
<td>100%</td>
<td>FINAL</td>
<td>Reporting System</td>
</tr>
<tr>
<td>DS</td>
<td>3</td>
<td>95%</td>
<td>FINAL</td>
<td></td>
</tr>
<tr>
<td>APC</td>
<td>4</td>
<td>93%</td>
<td>FINAL</td>
<td></td>
</tr>
<tr>
<td>ELCS</td>
<td>4</td>
<td>95%</td>
<td>FINAL</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>3</td>
<td>96%</td>
<td>FINAL</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>96%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E7 (US)</td>
<td>5</td>
<td>83%</td>
<td>FINAL</td>
<td></td>
</tr>
<tr>
<td>E7 (Eur)</td>
<td>1</td>
<td>97%</td>
<td>NEW</td>
<td></td>
</tr>
</tbody>
</table>

Source: Stuart Compton, Air Products plc

## Rolling DDP

### Software Testing Defect Detection Percentage Measure

(rolling quarterly produced values looking back four quarters)

<table>
<thead>
<tr>
<th>Period under review</th>
<th># Projects Analysed</th>
<th>Defects in Testing</th>
<th>Total Defects</th>
<th>Prod'n Bugs</th>
<th>DDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Historical Estimate</td>
<td>n/a</td>
<td></td>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>Rolling 1 Qtrs DDP to Q1 Y1</td>
<td>2</td>
<td>n/a</td>
<td>1111</td>
<td>1400</td>
<td>289</td>
</tr>
<tr>
<td>Rolling 2 Qtrs DDP to Q2 Y1</td>
<td>1</td>
<td>n/a</td>
<td>1171</td>
<td>1466</td>
<td>295</td>
</tr>
<tr>
<td>Rolling 3 Qtrs DDP to Q3 Y1</td>
<td>1</td>
<td>n/a</td>
<td>1211</td>
<td>1508</td>
<td>297</td>
</tr>
<tr>
<td>Rolling 4 Qtrs DDP to Q4 Y1</td>
<td>2</td>
<td>n/a</td>
<td>1492</td>
<td>1807</td>
<td>315</td>
</tr>
<tr>
<td>Rolling 4 Qtrs DDP to Q1 Y2</td>
<td>3</td>
<td>80</td>
<td>2034</td>
<td>2129</td>
<td>95</td>
</tr>
<tr>
<td>Rolling 4 Qtrs DDP to Q2 Y2</td>
<td>0</td>
<td>80</td>
<td>1974</td>
<td>2063</td>
<td>89</td>
</tr>
<tr>
<td>Rolling 4 Qtrs DDP to Q3 Y2</td>
<td>3</td>
<td>80</td>
<td>2086</td>
<td>2204</td>
<td>118</td>
</tr>
<tr>
<td>Rolling 4 Qtrs DDP to Q4Y2</td>
<td>2</td>
<td>80</td>
<td>1976</td>
<td>2087</td>
<td>111</td>
</tr>
<tr>
<td>Rolling 4 Qtrs DDP to Q1 Y3</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rolling 4 Qtrs DDP to Q2 Y3</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rolling 4 Qtrs DDP to Q3 Y3</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rolling 4 Qtrs DDP to Q4 Y3</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Stuart Compton, Air Products plc
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?

Contents

- **Introduction: some questions for you**
- **What is DDP**
- **Case studies**
- **Examples of calculating DDP**
- **Advice about DDP & Conclusion**
# DDP example

<table>
<thead>
<tr>
<th>Testing</th>
<th>Live Running</th>
<th>DDP after Live</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>50</td>
<td>75%</td>
</tr>
</tbody>
</table>

$$\text{DPD} = \frac{150}{150 + 50} = \frac{150}{200} = 75\%$$

## DDP is not percent of total defects

<table>
<thead>
<tr>
<th>Stage of testing</th>
<th>Defects</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Official&quot; (Mod &amp; Int)</td>
<td>299</td>
<td>75%</td>
</tr>
<tr>
<td>&quot;Tool&quot; &amp; development</td>
<td>40</td>
<td>10%</td>
</tr>
<tr>
<td>Release testing</td>
<td>19</td>
<td>5%</td>
</tr>
<tr>
<td>User Acceptance test</td>
<td>10</td>
<td>2.5%</td>
</tr>
<tr>
<td>Pilot</td>
<td>9</td>
<td>2.5%</td>
</tr>
<tr>
<td>Live use (1 mo.)</td>
<td>20</td>
<td>5%</td>
</tr>
</tbody>
</table>

This is not DDP (it’s %)
DDP compares testing processes

<table>
<thead>
<tr>
<th>Stage of testing</th>
<th>defects</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Official&quot; (Mod &amp; Int)</td>
<td>299</td>
<td>75%</td>
</tr>
<tr>
<td>&quot;Tool&quot; &amp; development</td>
<td>40</td>
<td>40%</td>
</tr>
<tr>
<td>Release testing</td>
<td>19</td>
<td>33%</td>
</tr>
<tr>
<td>User Acceptance test</td>
<td>10</td>
<td>25%</td>
</tr>
<tr>
<td>Pilot</td>
<td>9</td>
<td>31%</td>
</tr>
<tr>
<td>Live use (1 mo.)</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

live use determines DDP

DDP example: ST DDP after UAT

<table>
<thead>
<tr>
<th>ST</th>
<th>UAT</th>
<th>Live</th>
<th>ST DDP after UAT</th>
<th>ST DDP after Live</th>
<th>UAT DDP after Live</th>
<th>All test DDP after Live</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td></td>
<td>67%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DDP = \[
\frac{100}{100 + 50} = \frac{100}{150} = 67\%
\]
**DDP example: ST DDP after Live**

<table>
<thead>
<tr>
<th></th>
<th>ST</th>
<th>UAT</th>
<th>Live</th>
<th>ST DDP after UAT</th>
<th>ST DDP after Live</th>
<th>UAT DDP after Live</th>
<th>All test DDP after Live</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>100</td>
<td></td>
<td>67%</td>
<td>40%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{DDP} = \frac{100}{100 + 50 + 100} = \frac{100}{250} = 40\%
\]

**DDP example: UAT DDP after Live**

<table>
<thead>
<tr>
<th></th>
<th>ST</th>
<th>UAT</th>
<th>Live</th>
<th>ST DDP after UAT</th>
<th>ST DDP after Live</th>
<th>UAT DDP after Live</th>
<th>All test DDP after Live</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>100</td>
<td></td>
<td>67%</td>
<td>40%</td>
<td>33%</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{DDP} = \frac{50}{50 + 100} = \frac{50}{150} = 33\%
\]
DDP example: all test DDP after Live

<table>
<thead>
<tr>
<th></th>
<th>ST</th>
<th>UAT</th>
<th>Live</th>
<th>ST DDP after UAT</th>
<th>ST DDP after Live</th>
<th>UAT DDP after Live</th>
<th>All test DDP after Live</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>67%</td>
<td>40%</td>
<td>33%</td>
<td>60%</td>
</tr>
</tbody>
</table>

$$DDP = \frac{100 + 50}{100 + 50 + 100} = \frac{150}{250} = 60\%$$

Prediction of remaining faults

- DDP = 66%
- Faults found so far: 20
- Predicted faults not found yet: 10
- 50%
- Predicted faults not found yet: 20
- 80%
- Predicted faults not found yet: 5
Contents

- Introduction: some questions for you
- What is DDP
- Case studies
- Examples of calculating DDP
- Advice about DDP & Conclusion

Advice about DDP

- **What is not important**
  - projects of difference sizes, scope, duration, technology, life cycle approaches
  - how you collect and analyse defect data
  - really good accuracy or detailed classification (e.g. duplicates, where inserted)
  - how well the project actually went

- **What is important**
  - consistency of defect data collection and analysis
    - either use severity or not (same approach)

- **May matter / may not**
  - different people, different test approaches, complexity
DDP limitations

- not useful unless you have a reasonable number of defects and projects, over time
- never use it to measure individuals!
  - only test efforts
- don’t get sucked into too much detail (too soon)
  - don’t do hard things – do easy things
- it reflects what is happening
  - not [just] how good a job you are doing of testing
- not related to efficiency / cost

Technical aspects

- What time frame should I use for defects found in live?
  - this is arbitrary / whatever makes sense for you
  - many people use 1 month, some use 3 or 6 months
- Can I measure DDP of different test stages?
  - you can measure any stage as long as you have defects that came afterwards
    - but don’t measure individual people!!
- Can I use DDP in incremental / RAD development?
  - you have choices – accumulate, or measure until next release
- What if different defect tracking systems?
  - ok to combine for different stages or if consistently recorded
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

Accuracy of defect data

- **most common “stumbling block”**
  - what about duplicates?
  - what about enhancement requests?
  - what if some aren’t really defects?

- **the same answer always applies**
  - it doesn’t matter how you do it
  - as long as you do it the same way each time!

- **most useful aspect of DDP**
  - trends, changes over time
The technical person’s trap

- we’re testers – we can see all the problems!
  - you will think of lots of “problems” with this metric
  - yes, DDP (as any measure) can be mis-used
    • but that doesn’t mean it can’t be useful
- take the high level view
  - DDP, warts and all, computed simply and consistently, can help you monitor your testing processes
  - and show the effects of both good and bad things

When NOT to use DDP

- when you don’t have many defects
  - in test or in production (i.e. very high quality software)
- your defect tracking is immature, purely subjective, untrustworthy, or non-existent
- the software products you produce
  - are never used by anyone (no live running)
  - it doesn’t matter how many defects are in them
- it is impossible to get data on defects found in live running
  - (difficult is OK!)
- you’re not interested in improving
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
- can help predict future defect levels
- is applicable over different projects
  - reflects testing process in general

- can give on-going monitoring of testing

Summary: key points

- DDP requires counts of defects
  - but does not need great accuracy

- DDP is a useful measure
  - easy to calculate
  - based on defect data you probably already have
  - can tell you how effective your testing efforts are
    - and how other things affect it
DDP Exercise 3

If you have any project data from your own projects, put the numbers in the relevant columns. Calculate your own DDP using a calculator or using the workshop leader’s spreadsheet.

If you do not have real project data, put in your best guess from a previous project, or work with a neighbor on their figures.

Fault Information

<table>
<thead>
<tr>
<th>Release or project name</th>
<th>System Test or other test</th>
<th>UAT or other test stage</th>
<th>Live running (1 month)</th>
<th>ST DDP after UAT</th>
<th>ST DDP after LR</th>
<th>UAT DDP after LR</th>
<th>All test DDP after LR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How did you get these figures? (show your working out below or in the cells)
DDP Exercise 1

The following data has been recorded for a project.

Calculate the DDP of each testing stage based on all the defect information.

**Fault Information**

<table>
<thead>
<tr>
<th>Testing stage</th>
<th>Number of faults</th>
<th>DDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Official” testing – module and integration</td>
<td>299</td>
<td>= --- = --- = %</td>
</tr>
<tr>
<td>“Tool” testing &amp; development</td>
<td>40</td>
<td>= --- = --- = %</td>
</tr>
<tr>
<td>Release testing</td>
<td>19</td>
<td>= --- = --- = %</td>
</tr>
<tr>
<td>User Acceptance test</td>
<td>10</td>
<td>= --- = --- = %</td>
</tr>
<tr>
<td>Pilot</td>
<td>9</td>
<td>= --- = --- = %</td>
</tr>
<tr>
<td>Live Running (after one month)</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

Hint: you don’t really need a calculator – just round the numbers to the nearest 10 and you will be close enough!

\[
\text{DDP} = \frac{\text{Defects found in this stage of testing}}{\text{Defects found in this and all subsequent stages of testing}}
\]
## DDP Exercise 1 Solution: DDP Calculation

<table>
<thead>
<tr>
<th>Testing stage</th>
<th>No. faults</th>
<th>DDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) “Official” testing – module and integration</td>
<td>299</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>300 / 400</td>
</tr>
<tr>
<td>2) “Tool” testing &amp; development</td>
<td>40</td>
<td>40%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40 / 100</td>
</tr>
<tr>
<td>3) Release testing</td>
<td>19</td>
<td>33%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 / 60</td>
</tr>
<tr>
<td>4) User Acceptance test</td>
<td>10</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 / 40</td>
</tr>
<tr>
<td>5) Pilot</td>
<td>9</td>
<td>33% (or 31%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 / 30 (9 / 29)</td>
</tr>
<tr>
<td>6) Live Running (after one month)</td>
<td>20</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### How did we get these figures?

Remember DDP = Defects found in testing / all subsequent defects

#### Stage 1 “Official” testing
- Test stage 1 found approximately 300 defects – this is the numerator (top)
- The sum of all the subsequent stages is = 40 + 20 (rounded up) + 10 + 10 (rounded up) + 20 = 100
- So the denominator (bottom of the equation) is 300 + 100 = 400
- DDP for Stage 1 is therefore 300/400 or 75%

#### Stage 2 “Tool” testing
- Test stage 2 found 40 defects – this is the numerator (top)
- The sum of all the subsequent stages is = 20 (rounded up) + 10 + 10 (rounded up) + 20 = 60
- So the denominator (bottom of the equation) is 40 + 60 = 100
- DDP for Stage 2 is therefore 40/100 = 40%

#### Stage 3 Release testing
- Test stage 3 found 19 defects (round up to 20) – this is the numerator (top)
- The sum of all the subsequent stages is = 10 + 10 (rounded up) + 20 = 40
- So the denominator (bottom of the equation) is 20 + 40 = 60
- DDP for Stage 3 is therefore 20/60 = 33%

#### Stage 4 User Acceptance test
- Test stage 4 found 10 defects – this is the numerator (top)
- The sum of all the subsequent stages is = 10 (rounded up) + 20 = 30 (29 to be exact)
- So the denominator (bottom of the equation) is 10 + 30 = 40
- DDP for Stage 4 is therefore 10/40 = 25%

#### Stage 5 Pilot
- Test stage 5 found 9 defects (round up to 10) – this is the numerator (top)
- The sum of all the subsequent stages is = 20 (the only remaining stage is live running)
- So the denominator (bottom of the equation) is 10 + 20 = 30
- DDP for Stage 4 is therefore 10/30 = 33% (31% if you calculate 9/29)

There is no DDP for live running, since the live running total goes into the calculation of all the previous DDP’s.
DDP Exercise 2

The following data has been recorded for a project.

Calculate the DDP’s in the columns on the right. The first one has been done as an example. (A calculator may be useful for some of these – your mobile phone has one!)

**Fault Information**

<table>
<thead>
<tr>
<th>Release</th>
<th>System Test</th>
<th>User Acceptance Test</th>
<th>Live running (1 month)</th>
<th>ST DDP after UAT</th>
<th>ST DDP after LR</th>
<th>UAT DDP after LR</th>
<th>All test DDP after LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 1</td>
<td>100</td>
<td>50</td>
<td>100</td>
<td>67%</td>
<td>40%</td>
<td>33%</td>
<td>60%</td>
</tr>
<tr>
<td>Release 2</td>
<td>150</td>
<td>50</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release 3</td>
<td>200</td>
<td>50</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release 4</td>
<td>50</td>
<td>25</td>
<td>125</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**How did we get these figures?**

Release 1

ST DDP after UAT: 100 / (100 + 50) = 100 / 150 = 67%
ST DDP after LR: 100 / (100 + 50 + 100) = 100 / 250 = 40%
UAT DDP after LR: 50 / (50 + 100) = 50 / 150 = 33%  (Remember not to include the ST defects here)
All test DDP after LR: (100 + 50) / (100 + 50 + 100) = 150 / 250 = 60%
DDP Exercise 2 Solution: DDP Calculations

<table>
<thead>
<tr>
<th>Release</th>
<th>System Test</th>
<th>User Acceptance Test</th>
<th>Live running (1 month)</th>
<th>ST DDP after UAT</th>
<th>ST DDP after LR</th>
<th>UAT DDP after LR</th>
<th>All test DDP after LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 1</td>
<td>100</td>
<td>50</td>
<td>100</td>
<td>67%</td>
<td>40%</td>
<td>33%</td>
<td>60%</td>
</tr>
<tr>
<td>Release 2</td>
<td>150</td>
<td>50</td>
<td>10</td>
<td>75%</td>
<td>71%</td>
<td>83%</td>
<td>95%</td>
</tr>
<tr>
<td>Release 3</td>
<td>200</td>
<td>50</td>
<td>50</td>
<td>80%</td>
<td>67%</td>
<td>50%</td>
<td>83%</td>
</tr>
<tr>
<td>Release 4</td>
<td>50</td>
<td>25</td>
<td>125</td>
<td>67%</td>
<td>25%</td>
<td>17%</td>
<td>38%</td>
</tr>
</tbody>
</table>

**How did we get these figures?**

**Release 2**
- ST DDP after UAT: $\frac{150}{150 + 50} = \frac{150}{200} = 75\%$
- ST DDP after LR: $\frac{150}{150 + 50 + 10} = \frac{150}{210} = 71\%$
- UAT DDP after LR: $\frac{50}{50 + 10} = \frac{50}{60} = 83\%$
- All test DDP after LR: $\frac{150 + 50}{150 + 50 + 10} = \frac{200}{210} = 95\%$

**Release 3**
- ST DDP after UAT: $\frac{200}{200 + 50} = \frac{200}{250} = 80\%$
- ST DDP after LR: $\frac{200}{200 + 50 + 50} = \frac{200}{300} = 67\%$
- UAT DDP after LR: $\frac{50}{50 + 50} = \frac{50}{100} = 50\%$
- All test DDP after LR: $\frac{200 + 50}{200 + 50 + 50} = \frac{250}{300} = 83\%$

**Release 4**
- ST DDP after UAT: $\frac{50}{50 + 25} = \frac{50}{75} = 67\%$
- ST DDP after LR: $\frac{50}{50 + 25 + 125} = \frac{50}{200} = 25\%$
- UAT DDP after LR: $\frac{25}{25 + 125} = \frac{25}{150} = 17\%$
- All test DDP after LR: $\frac{50 + 25}{50 + 25 + 125} = \frac{75}{200} = 38\%$