→ www.steria.co.uk

Defect Density Measurement

Peter Thomas CITP CFPS
2011

Contact peter3292004@hotmail.co.uk

Steria is a multi national European company which does about One Billion Euros of Services and other IT business each year. See WWW.STERIA.COM for more details.

Executive Summary – “What’s in it for me”

- To enable process and efficiency improvements, current performance must be measured
- Counting Defects is misleading but Defect Density, the ratio of defects to size, is a recognised industry standard
- The organisation can use analysis and reporting to track trends, identify outliers, and trigger Process Improvements
- Optional Benchmark performance against the industry can be carried out (eg with Gartner, Compass, ISBSG).

Defect - The lack of something necessary or desirable for completion or perfection – discussed in more detail later

Defect Count v Defect Density

- Counting defects does not give management information – and can be misleading
- See example in table where the biggest defect count is actually the best quality by an order of magnitude and the quality of the other two projects is equal
- Industry Best Practice is to measure the Defect Density as the quality indicator
- Other measurements are required to verify causal analysis and process behaviour

Release 1 is a small enhancement	**Release 2 is new set of extracts**	**Release 3 is new third party software**
20 FP*	100 FP	1000 FP
2 defects	10 defects	20 defects
Defect Density (per 100 FP) = 10	Defect Density (per 100 FP) = 10	Defect Density (per 100 FP) = 2

*FP = Function Point

Defect Density Definition

- Defect is defined by ANSI/IEEE Std 729-1983 defines a [defect] as,
  - "The termination of the ability of a functional unit to perform its required function.“
- Defect Density is a measure showing the ratio of defects against the size of a development (Number of Defects/size). Size is typically expressed in terms of Function Points (FP), Impact Points or other ‘points’ measures
- It is normally reported as defects per 100 or 1000 ‘points’
  - For example, acceptable delivered quality is less than 1 severity 1 defect, during a 90 day warranty period following delivery into production, per 100 FP

Note – the “first computer bug” is here
http://americanhistory.si.edu/collections/comphist/objects/bug.htm

How does defect density benefit the organisation?

- Enables the quality of similar projects to be compared (this cannot be achieved by simply counting defects)
- Testing effectiveness is measured by a decrease in defect density in the subsequent test phases (scope may have changed)
- Trend analysis of Defect Density can be used to demonstrate improvements;
  - An overall reduction in defects can be used to demonstrate an increase in the overall quality of the delivered product
  - A reduction in defects in “integration in the small” phase would show an improvement in quality in the build activities
- Can be used to improve estimation by providing historical defect density data on projects with similar size/profiles
- Optionally can be used to benchmark the organisation against similar organisations

Possible use in Estimating

- The measurement repository will contain historical data on:
  - Size (Fast Function/Impact/Coverage points value)
  - Complexity (questionnaire based assessment)
  - Defect Density
  - Other project profile information (eg Project Type) plus names of managers involved
- By searching for projects with similar profiles this data could be used, in combination with historical estimation MI data, to improve the estimating process
  - Historical project data could be used to ratify the feasibility phase estimates
  - Once the project commences early sizing assessments could be undertaken as part of top down Estimates to verify the bottom up task (WBS) based approach to estimating
  - This could be repeated as part of detailed estimating as more information becomes available (exit plan phase)
  - Each projects data will be added to the measurement repository as it progresses to further enhance the historical data for future projects

Filtering Incidents to get defects

- Although items entered into the “Defect” data store are called defects, in reality they are “incidents” which are one of the following:
  - Change Request Agreed
  - Deferred
  - Duplicate
  - Existing Production Incident
  - Merged with another Defect
  - No Longer an Issue
  - Not a fault
  - Not in Scope of Project
  - Resolution Implemented
    - “Our problem”
    - “some-one else”
  - Referred to another project
  - Third Party Fix
  - Risk accepted by the business
  - Workaround accepted by the business

Options for Sizing

- Based on International Standards for Functional Size Measurement (FSM) ISO/IEC 14143-1:1998
  - IFPUG FPs (ISO/IEC 20926:2010)
- Based on variant to include non functional requirements
  - Impact Points
- Based on other somewhat recognised standards
  - Feature Points etc
- Other
  - Lines of Code (LOCs)
  - Questionnaire based scorecard

These are reviewed in more detail on the following slides.

Note: Development effort (hours/days) is not a reliable indicator of product size.

Function Points (FP)

- Method has been in use for around 30 years
  - Introduced to overcome issues with other sizing methods eg LOCs
- The project functional size uses the introduction, modification or removal of business functionality as input
  - Identify and count internal data stores, external data stores, inputs, queries, outputs
- Pros
  - Detailed sizing method available (200 plus page Counting Practices Manual plus case studies etc.) which ensures consistent results
  - Most industry data is based on FP
  - Can be performed at a High level (‘Fast FPs’ eg FP Lite developed by DCG) for a good indication of size
- Cons
  - The project functional size may not cover all of the scope which may give rise to defects
  - Can be expensive to count (if performed to IFPUG standard)

Impact Points

- Extends the formal FPA method by including components that are changed without changing business functionality (non functional requirements eg performance)
- Pros
  - Projects can be sized using a mix of FPs and Impact Points to cover the different scenarios to derive an aggregate Impact Point size
- Cons
  - Cannot easily be benchmarked against the industry

Note: This reference provides more details of the method http://www.qpmg.com/pdf/counting-practices/Impact Points Counting Guidelines - v1.pdf

Feature Points

- Extension of Function Points with added parameter to measure the number of algorithms
- Proprietary to Software Productivity Research(SPR) (http://www.spr.com/function-points-or-feature-points.html)
- Pros
  - Differentiates between typical MIS products and premium calculation and other more sophisticated applications
- Cons
  - Still ignores many non functional requirements which are causing code introduction and modification, hence defects
  - More complex to derive than other ‘points’ measures

Lines of Code

- Counting tools have own definition of a line of code
- Pros
  - Relatively easy to count assuming automated tool installed (otherwise labour intensive and prone to inconsistencies)
- Cons
  - Meaningless to the business user
  - Easy for projects to add more code than necessary to improve the metrics
  - Comparison between code languages is meaningless
  - “Back-firing” – conversion to FP is not valid – method now discounted by the author Capers Jones
  - Source code for third party applications/packages not available
  - Not applicable to Infrastructure projects

Project Sizing Assessment (S/M/C)

- This would be a sizing method developed specifically for the organisation, consisting of a number of factors that are perceived to impact project size. A checklist would need to be produced using the sizing factors and value ranges to provide a numerical size value.
- Pros
  - Simply/straightforward to use
  - Allows project to be assigned to category eg 1-5 range
- Cons
  - Does not provide a measurement to allow defect density to be measured. It is a statistical ordinal measurement which only allows ranking not measurement
  - Completion of checklist questions subjective therefore best performed by sizing “expert” to ensure consistency
  - Unproven, would need regular monitoring/re-calibration
  - Cannot be benchmarked against the industry

Infrastructure sizing

Infrastructure Project categories:

- ➔ Change of middleware, for example CICS
  - "middleware" is a layer of software between the user and the hardware.
  - "application" is the layer of software with which the user interacts and contains the business rules

- ➔ Change of COTS package, for example Lotus Notes
  - ➔ “package” provides business function using a combination of:
    - ➔ As is code
    - ➔ Tailored / configured function
    - ➔ “bolt on” / user exit code eg SAP ABAP

- ➔ Infrastructure projects will be sized in 2 ways:
  - ➔ Coverage points for tested application/area
  - ➔ Impact points for changed application code (non-functional)
  - ➔ Up to 2 Defect Density measures will be produced and defects will be allocated by Resolver Group

Change of middleware

- Middleware is not directly tested. Scope of change is unknown to purchasing organisation
- Installation performed by following / executing script[s]
  - Not measurable
- Smoke and/or regression and/or penetration tests are undertaken on application[s] or parts of application (including packages) that use the middleware
  - This is measured using ‘Coverage points’
    - These will follow the same rules as for counting Function/Impact Points
    - The Coverage points for a whole application will be the Function point size of the application.
- Application code change may be required, may be measured with component and functional testing
  - The changed code will be measured using Impact Points

Change of COTS package

- Installation performed by following / executing script[s]
  - Not measurable
- Smoke and or regression and or penetration tests are undertaken on package or parts of package
  - This is measured using ‘Coverage points’
    - See previous slide
- “bolt on” code or tailor/configuration change may be required, may be measured with component and functional testing
  - this is measured using Impact Points

Organisation Project Profiles

Function/Impact/Coverage Points cover the majority of defects / projects at a typical software development organisation

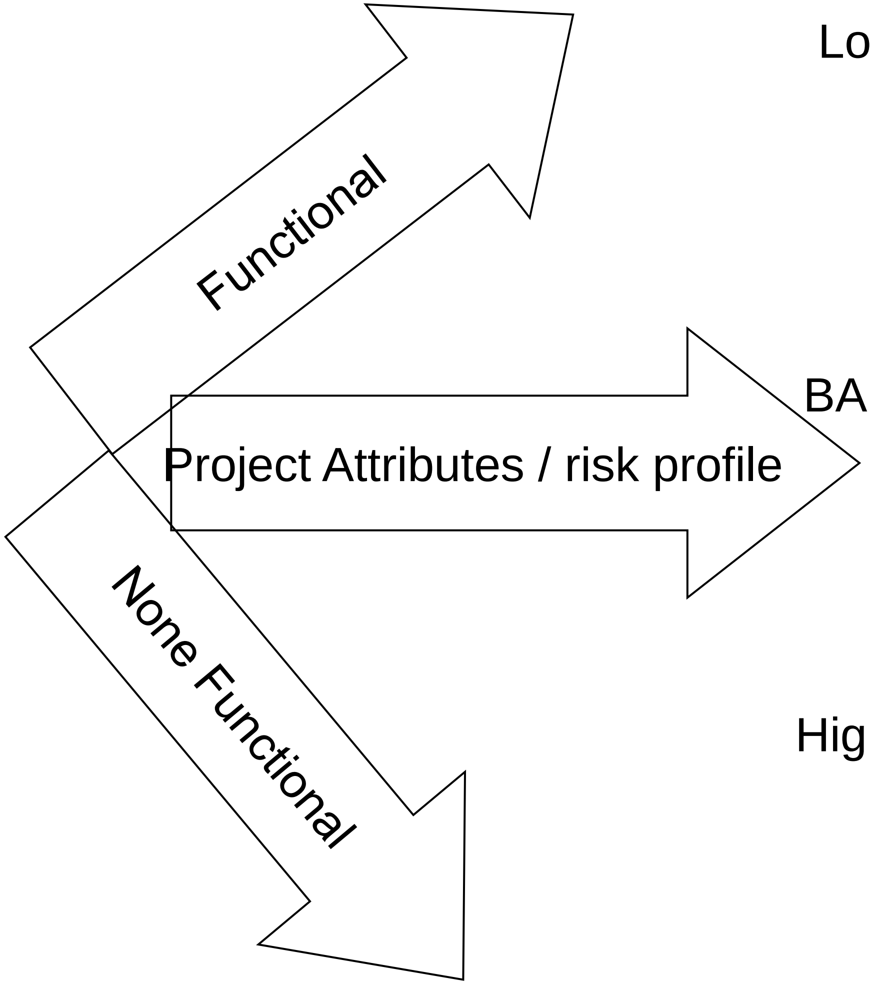
Project Category	Recommended Size Measure
Bespoke New Software	Function/Impact Points *
Amendment to Bespoke Software	Function/Impact Points *
New Third Party Package Implementation	Function/Impact Points *
Upgrade to Third Party Package	Function/Impact Points *
Migration of functionality to new systems (eg migration of life systems to Capita	Function/Impact Points *
New/Upgraded Infrastructure – networks/Firewalls/Routers	Coverage Points
New/Upgraded Hardware	Coverage Points
New/Upgraded firmware (Operating systems/Message Broker/CICS/VOIP etc)	Coverage Points
New/Upgraded Global packages (Browsers/Microsoft Office etc)	Coverage Points
Disaster Recovery Provision	Coverage Points

* Size Functional Requirements using Function Points. Size the other Requirements using Impact points and combine.

Cost of Accuracy in Sizing Projects

- Auditable sizing to IFPUG standards
  - Time consuming
  - Expensive
  - Estimate 2% of overall project size
- Fast Function/Impact Point counting
  - high level quick sizing (range of values)
  - Sizing effort time-boxed
  - Will provide ‘adequate’ measure if done with appropriately skilled resource
  - Estimate ½ day to less than 1% of overall project size
- Coverage Point counting
  - high level quick sizing (range of values)
  - Sizing effort time-boxed
  - Will provide ‘adequate’ measure if done with appropriately skilled resource
  - Estimate 1 day to less than 2% of overall project size

Defects to FP typically varies ??? To ??? per FP Will be outside that range for some project attributes



Functional

Project Attributes / risk profile

None Functional

Low defect density

- Small functional
- Easy none functional
- Easy / good project attributes

BAU defect density

- BAU functional
- BAU none functional
- BAU project attributes

High defect density

- High functional
- Hard none functional
- Difficult / poor project attributes

Surface joining points of equal defect density is a complex shape

Contextual Data

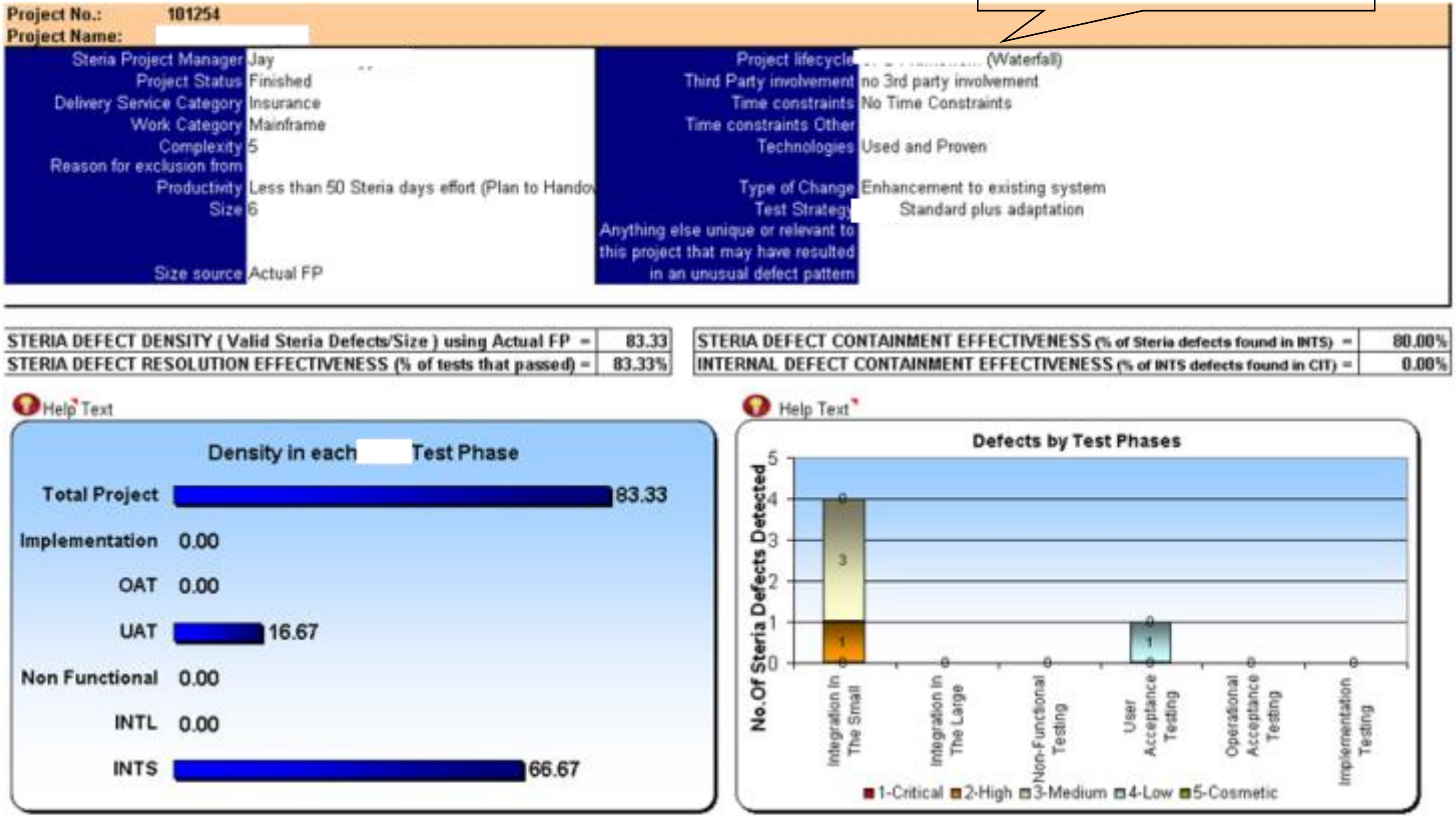
- As the defect measures are being applied to many types of project, it is necessary to capture a project profile to provide contextual data for the measures – reduce impact of non functional and project attribute dimension
- These could include (but not limited to):
  - Technology / infrastructure
  - Complexity (size algorithm - score based on several factors)
  - Project Type
    - New development
    - Enhancement
    - Package implementation/upgrade
    - Data migration
  - Project time constraints (anybody can deliver rubbish quickly)
- This data enables similar projects to be grouped (logically organized hierarchically in multiple dimensions aka slice and dice) and unusual defect patterns to be identified (split the basket of fruit into apples and pears)

ISBSG

- The latest Special Analysis Report on Software Defect Density from the ISBSG reveals useful information about defects in software, both in development and in the initial period after a system has gone into operation:
  - The split of where defects are found, i.e. in development or in operation, seems to follow the 80:20 rule. Roughly 80% of defects are found during development, leaving 20% to be found in the first weeks of systems' operation.
  - Fortunately, in the case of extreme defects, less than 2.5% were found in the first weeks of systems' operation.
  - Extreme defects make up only 2% of the defects found in the Build, Test and Implement tasks of software development.
  - The industry hasn't improved over time. Software defect densities show no changing trend over the last 15 years.
- The report also provides useful ratios of when and where defects are discovered and the severity of those defects. The analysis data set is very broad with over 30 different organization types represented and a hundred different application types.
- Analysis results are provided by:
  - Organization type;
  - Application type;
  - Development Platform;
  - Language Type;
  - Project Size; and
  - Speed of Delivery.
- The report can be used to benchmark against or to help with test planning and management.
- **http://isbsg.org/isbsgnew.nsf/WebPages/7487DAFCD90E2CA0CA257642000FA364**

Example report 1

Project details



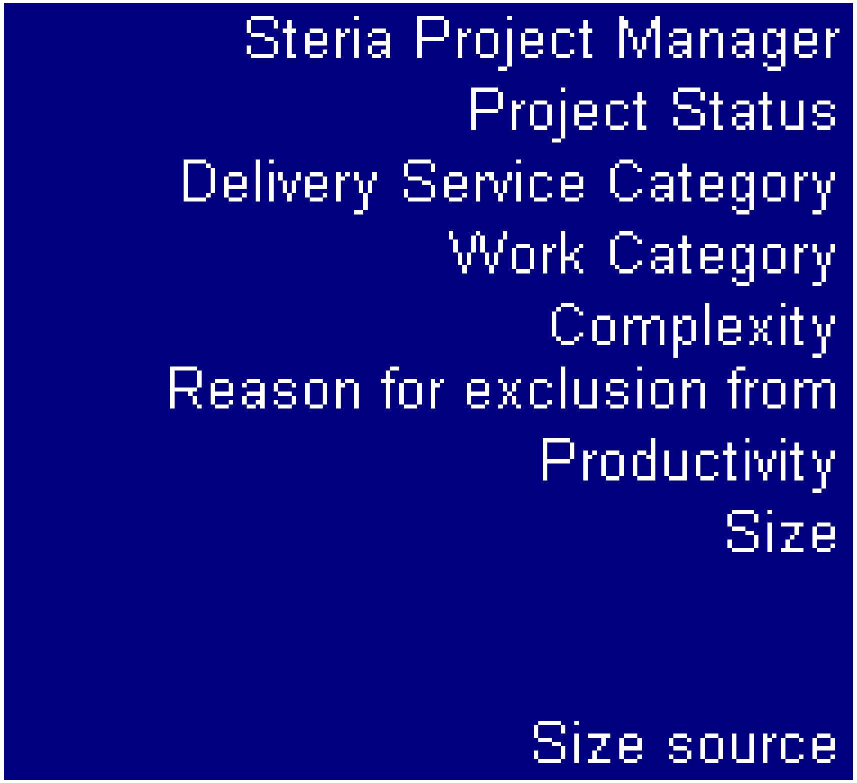
Project No.:	101254		
Project Name:			
Steria Project Manager	Jay	Project lifecycle	(Waterfall)
Project Status	Finished	Third Party involvement	no 3rd party involvement
Delivery Service Category	Insurance	Time constraints	No Time Constraints
Work Category	Mainframe	Time constraints Other	
Complexity	5	Technologies	Used and Proven
Reason for exclusion from Productivity	Less than 50 Steria days effort (Plan to Handov	Type of Change	Enhancement to existing system
Size	6	Test Strategy	Standard plus adaptation
Size source	Actual FP	Anything else unique or relevant to this project that may have resulted in an unusual defect pattern	

STERIA DEFECT DENSITY (Valid Steria Defects/Size) using Actual FP =	83.33	STERIA DEFECT CONTAINMENT EFFECTIVENESS (% of Steria defects found in INTS) =	80.00%
STERIA DEFECT RESOLUTION EFFECTIVENESS (% of tests that passed) =	83.33%	INTERNAL DEFECT CONTAINMENT EFFECTIVENESS (% of INTS defects found in CIT) =	0.00%

Density measurements

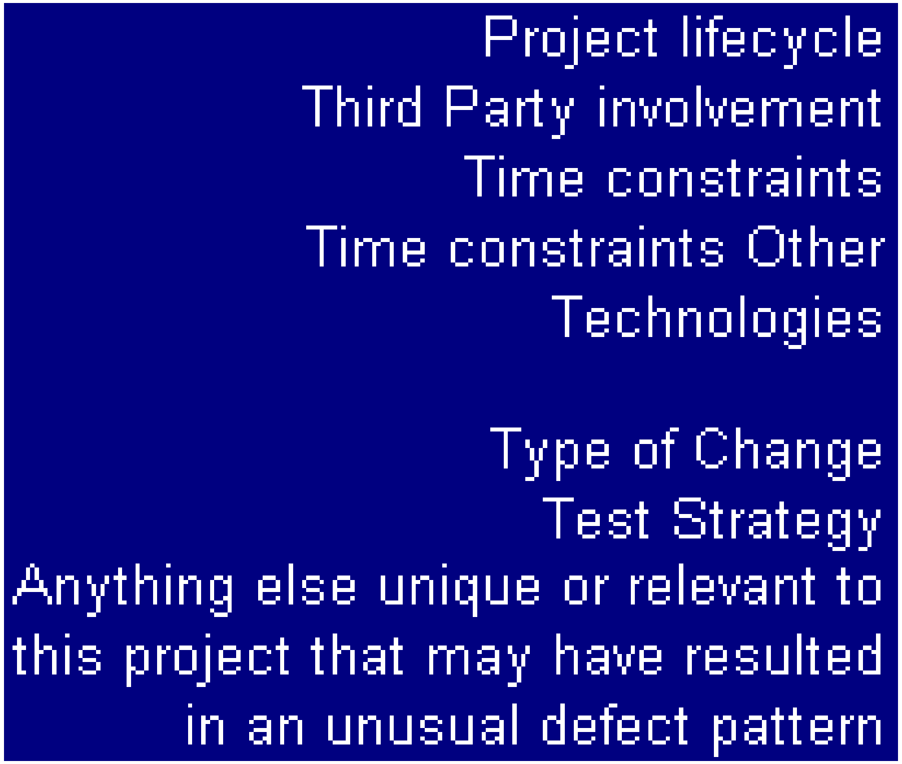
Phase distribution

Example report 1 – project details



Steria Project Manager
Project Status
Delivery Service Category
Work Category
Complexity
Reason for exclusion from Productivity
Size

Size source

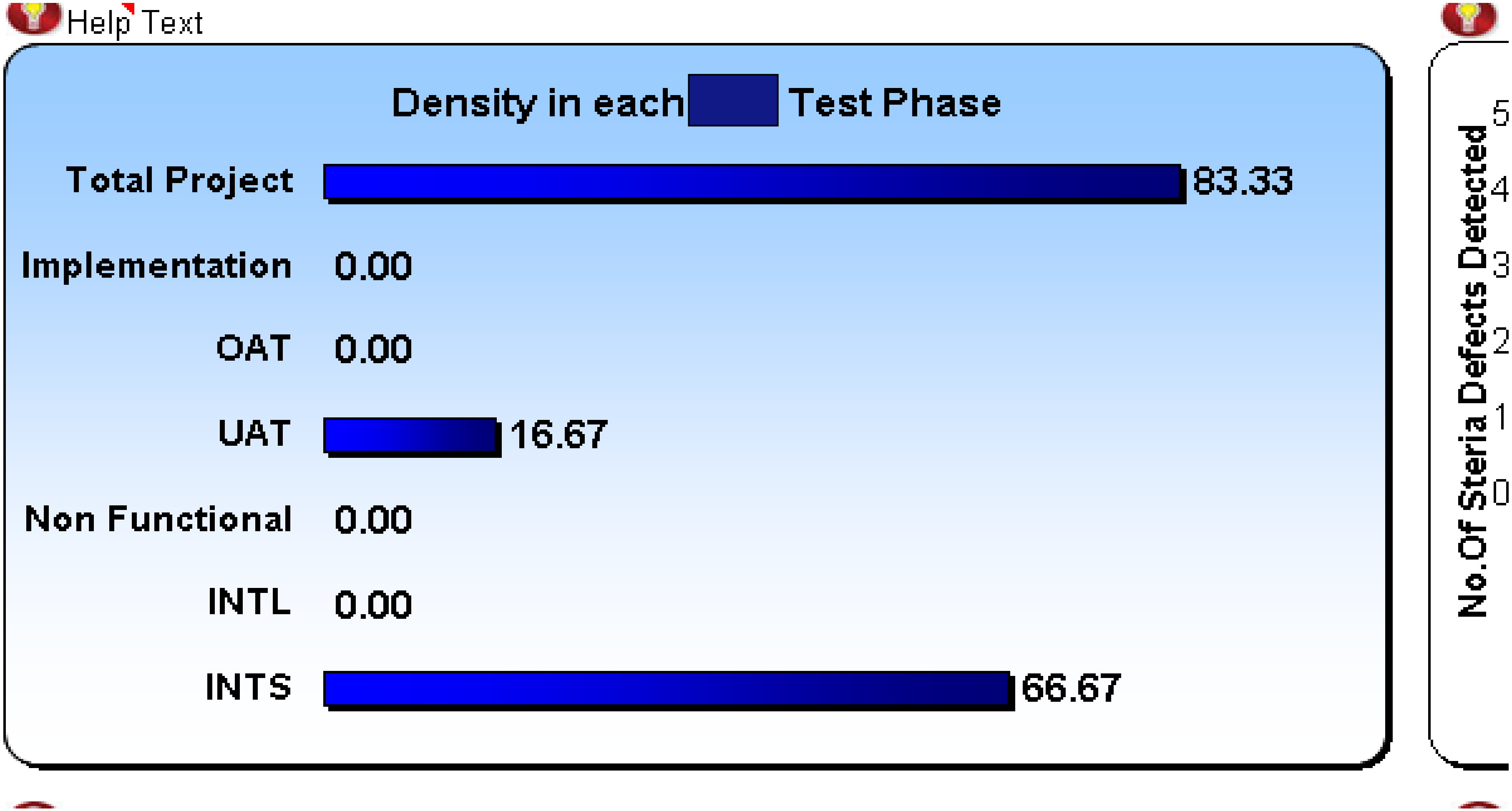


Project lifecycle
Third Party involvement
Time constraints
Time constraints Other
Technologies

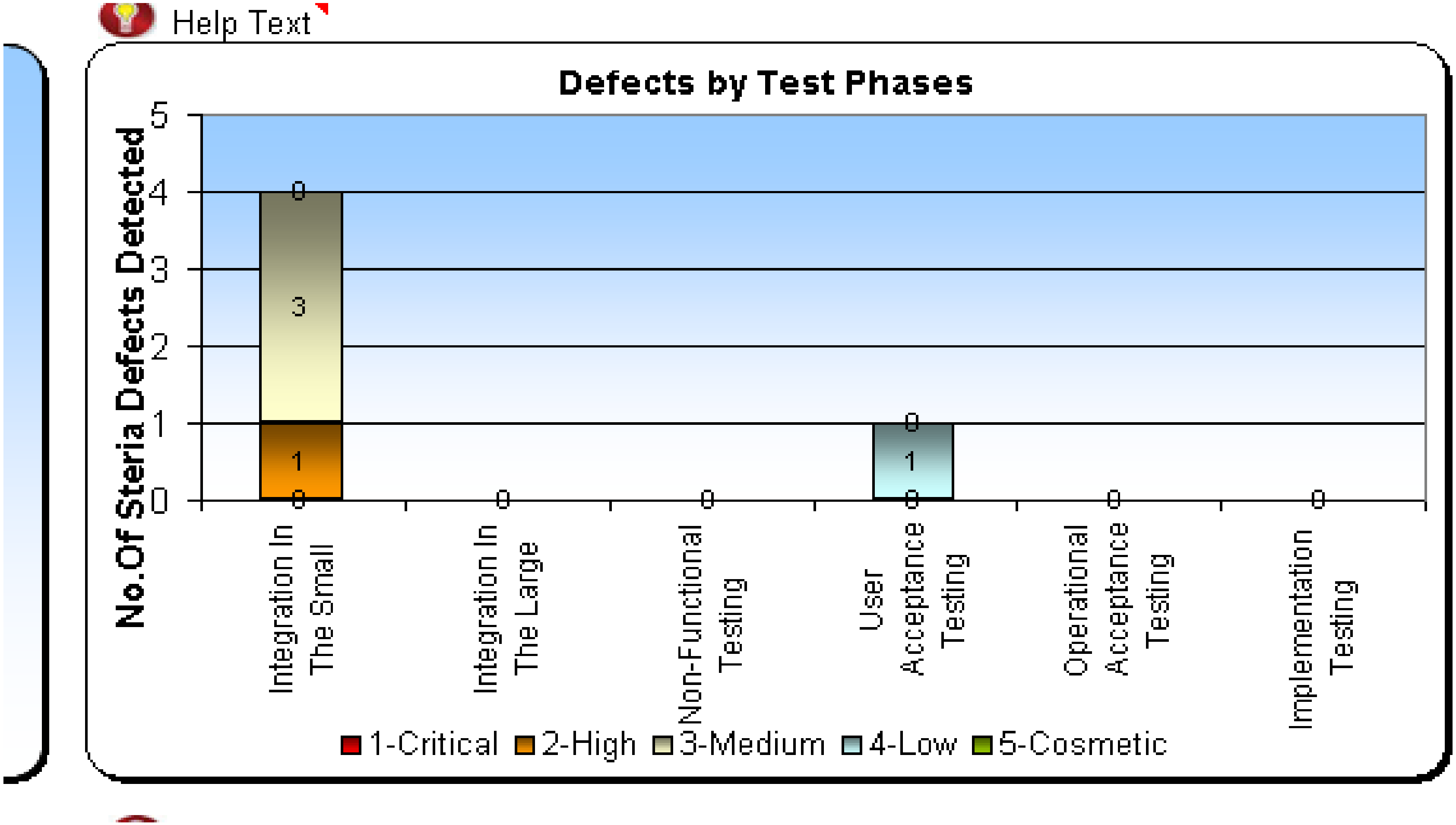
Type of Change
Test Strategy
Anything else unique or relevant to this project that may have resulted in an unusual defect pattern

These details retrieved from attributes in the project record in the data repository

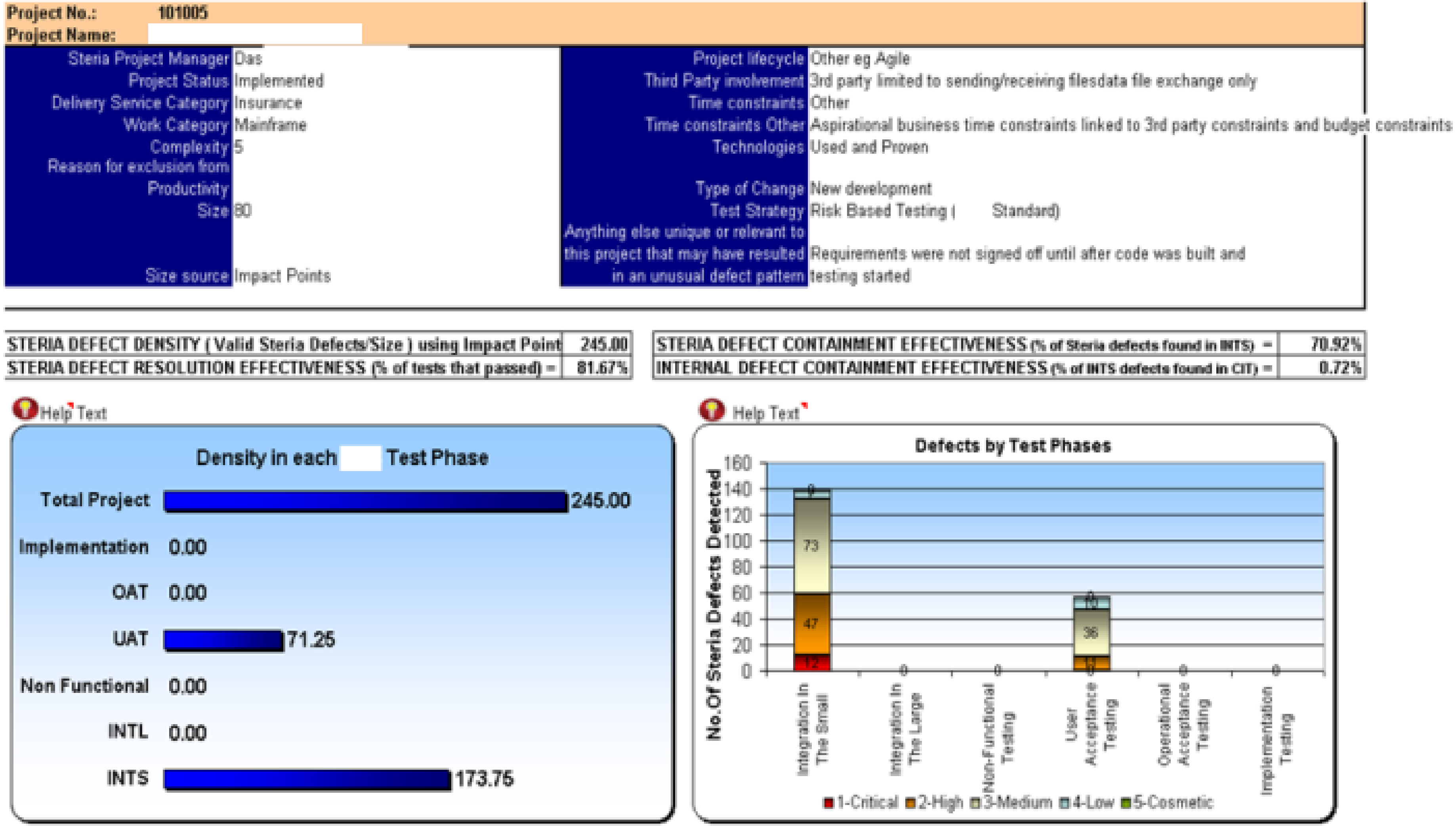
Example report 1 – Density Measurements



Example report 1 – Phase distribution



Example report 2



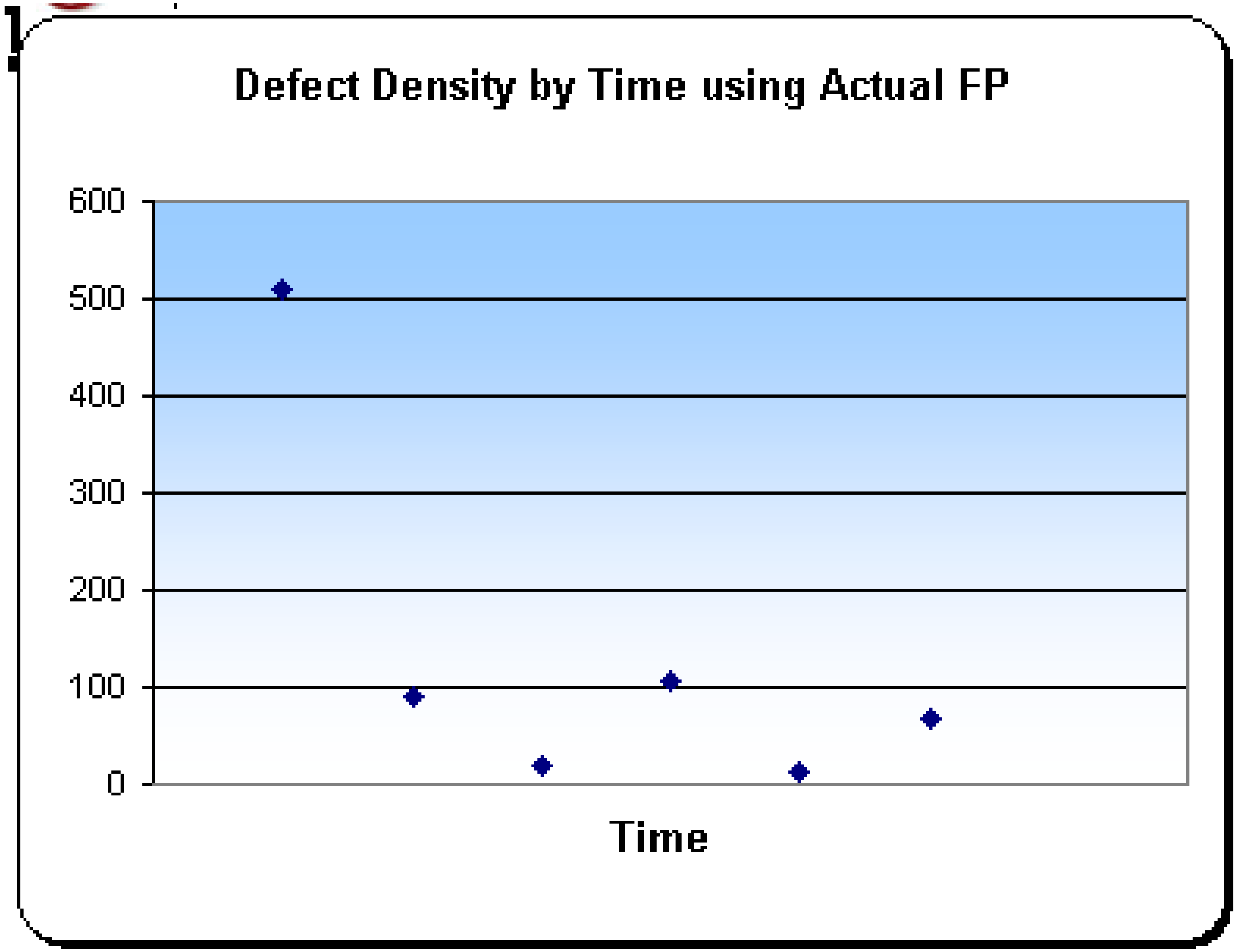
Project No.:	101005
Project Name:	

Steria Project Manager	Das	Project lifecycle	Other eg Agile
Project Status	Implemented	Third Party involvement	3rd party limited to sending/receiving filesdata file exchange only
Delivery Service Category	Insurance	Time constraints	Other
Work Category	Mainframe	Time constraints Other	Aspirational business time constraints linked to 3rd party constraints and budget constraints
Complexity	5	Technologies	Used and Proven
Reason for exclusion from Productivity		Type of Change	New development
Size	80	Test Strategy	Risk Based Testing (Standard)
Size source	Impact Points	Anything else unique or relevant to this project that may have resulted in an unusual defect pattern	Requirements were not signed off until after code was built and testing started

STERIA DEFECT DENSITY (Valid Steria Defects/Size) using Impact Point	245.00
STERIA DEFECT RESOLUTION EFFECTIVENESS (% of tests that passed) =	81.67%

STERIA DEFECT CONTAINMENT EFFECTIVENESS (% of Steria defects found in INTS) =	70.92%
INTERNAL DEFECT CONTAINMENT EFFECTIVENESS (% of INTS defects found in CIT) =	0.72%

Example report 3



Summary

- Counting Defects is misleading but Defect Density, the ratio of defects to size, is a recognised industry standard
- The organisation can use analysis and reporting to track trends, identify outliers, and trigger Process Improvements
- We have reviewed key messages that can be shared with executives to get their buy in to support defect density measurement.
- There are several candidates for the size measure which could be used for the defect density measurement. Function Points is the preferred measure.
- Non functional projects require other measures.
- Functional size, non functional requirements, and project attributes combine to give a expected hence outlier defect density.
- ISBSG and other repositories have some measurements
- Steria is capturing and reporting the measurement
- Predictive model is still a long way off